

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA



PIQL – Interfaces Persuasivas para uma Vida Saudável

Diogo Miguel Branco Silva

Mestrado em Engenharia Informática
Sistemas de Informação

Dissertação Orientada por:
Prof. Doutor Carlos Alberto Pacheco dos Anjos Duarte
José Romana Baptista Coelho

2017

Agradecimentos

Gostaria de agradecer ao meu orientador, Professor Carlos Duarte, bem como ao meu coorientador, José Coelho, por todos os conhecimentos que me transmitiram, não só a nível profissional, mas também a nível pessoal. Além disso, não posso deixar de referir todas as reuniões semanais, onde debatemos até ao mais ínfimo detalhe cada etapa a ser realizada futuramente neste projeto e como este poderia ser melhorado; todos os *emails*, desde as perguntas mais parvas que realizava até aos *updates* diários e todo o tempo que despenderam em prol deste trabalho, abdicando muitas vezes do vosso próprio. Por tudo o que já foi acima mencionado e por outros pontos que me possa ter esquecido, um obrigado não é suficiente.

Aos parceiros europeus, e especialmente ao psicólogo Jan Nordvik, pelo tremendo contributo prestado através da partilha de conhecimentos e aplicações, as quais foram essenciais para o desenvolvimento deste projeto.

À psicóloga Cristina Godinho, pelo contributo na validação de componentes e sugestões dadas para o desenvolvimento do projeto, o qual deu um maior valor ao mesmo.

Aos meus pais, Ilídia Silva e José Silva, que desde muito cedo me incutiram a trabalhar e a ser “alguém” na vida, por me terem ensinado que uma pessoa pode não ter o maior talento no mundo, mas se essa pessoa se aplicar e não desistir, consegue chegar ao seu objetivo. Por tudo o que já passaram na vida e por investirem da forma como investiram, e ainda investem, em mim, sinto-me orgulhoso dos pais que tenho, onde uma vez mais, um obrigado não é suficiente. Quero também agradecer a todos os meus familiares que de alguma forma me ajudaram a nível pessoal.

À minha namorada, Carolina Condeço, por ter estado sempre presente nesta etapa da minha vida, por ouvir sempre os meus desabafos, por ter constantemente uma palavra de apoio sempre que me via em baixo, pela dedicação e empenho que continuamente exibiu, e ainda exhibe, dia-a-dia desde o início da nossa relação, não só a nível pessoal, mas também a nível profissional e amoroso. Por tudo o que foi acima referido, fez-me querer tornar não só uma melhor pessoa em todos esses campos, mas também desejar um dia conseguir retribuir tudo o que já me deu e continua a dar.

Aos meus amigos de 2011 (Desmarques, DaCosta, Valeta, Maias, Floribela, GPS, Copy, Paste) os quais me apoiaram muitas vezes em inúmeras ocasiões, especialmente os

mais chegados. A estes últimos, para além de um “agradecimento”, um pedido de desculpas, por muitas vezes não ter tido a capacidade de estar sempre com vocês e priorizar aspetos da minha vida pessoal. É um aspeto que tenho a melhorar, não só para me tornar uma pessoa melhor, mas porque merecem ser reconhecidos pelo quanto me ajudam, mas também pelo quão essenciais vocês foram e continuam a ser.

Aos meus Padrinhos (Mafalda e Adultério). Por todo o conhecimento e carinho que me deram, por terem fundado as bases para o meu sucesso académico e por serem sempre um porto de abrigo, desde que vos conheci.

À minha outra Grande Família. Afilhados, netos, irmãos, tios, sobrinhos, primos, não interessa o estatuto, interessa o quão importantes foram e continuam a ser para mim. Cada um diferente à sua forma, mas todos em conjunto preenchem um vazio que nem sempre a minha família preenche. Obrigado a todos, especialmente aos meus afilhados os quais devo muito daquilo que sou, e um agradecimento ao Guilherme Espada (Andúril) pela disponibilidade e auxílio prestado neste trabalho.

Aos meus colegas de Mestrado, por tudo o que passamos juntos, quer em conhecimento, quer em amizade e também pelo facto de que nem sempre vos vejo presentemente, mas sempre que estamos juntos contagiam-me com a vossa boa atitude e disposição, trazendo consequentemente um sorriso ao meu rosto.

Aos amigos de Azambuja (Carvalho, Pedro, Serra, Tiago, Caca, Sara, Rebeca, Rita), um grande obrigado pela influência que tiveram, e que alguns ainda têm, em mim e consequentemente nesta Tese. Vocês são grandes.

Um pequeno agradecimento à minha pequena sobrinha Alice, que da sua própria forma, conseguia sempre alegrar as minhas Terças e Quintas.

Por último, às pessoas que não estão referenciadas diretamente em nenhuma destas secções (Faneca, Sargeta, Ana Santos, entre outras que me possam estar a escapar), parte do meu sucesso deve-se a vocês também e vocês sabem isso. Obrigado.

Resumo

O aumento da esperança média de vida não é acompanhada por um aumento da qualidade de vida, criando a necessidade de mecanismos que permitam esse acompanhamento. O projeto *PIQL* (*Persuasive Interfaces for Healthier Quality of Life*), como o próprio nome indica, visa a construção de interfaces persuasivas para uma melhor qualidade de vida. De acordo com esse objetivo houve a necessidade de identificar interfaces persuasivas existentes, criar um sistema persuasivo e integrar esse sistema no *PersonAAL*. O *PersonAAL* tem como objetivo construir mecanismos que permitam a aplicações web, focadas em idosos, serem aperfeiçoadas com mecanismos de adaptação e persuasão. Após a leitura de trabalho relacionado com interfaces persuasivas, modelos de comportamento e técnicas de alteração comportamentais, procedeu-se à implementação do sistema *PIQL*. Este consiste na criação de mensagens persuasivas com o intuito de incentivar o utilizador a melhorar o seu comportamento ou preservá-lo caso este já seja positivo. Esta implementação foi enquadrada no objetivo do *PersonAAL*. Ponderado o trabalho realizado e as propostas futuras de trabalho que o sistema *PIQL* possui, considera-se que os objetivos do projeto *PIQL* foram cumpridos com sucesso, existindo atualmente um sistema persuasivo integrado no *PersonAAL* validado através de vários tipos de testes, tendo sido a avaliação geral dos testes muito satisfatória. Embora os resultados sejam satisfatórios, o *PIQL* pode possuir novas funcionalidades que melhorem o sistema.

Palavras-chave: *PIQL*, Interfaces Persuasivas, Sistema de Regras, Persuasão, Qualidade de Vida.

Abstract

The increase of life expectancy is not followed by a growth of quality of life, creating the need of mechanisms to support it. *PIQL* (Persuasive Interfaces for Healthier Quality of Life) as the name itself might suggest, aims at the development of persuasive interfaces pointing at a better quality of life. According to that goal, the identification of existent persuasive interfaces was needed to create a persuasive system and integrate that system on *PersonAAL*. *PersonAAL* allows web applications, focused on older people, to be improved by persuasive and adaptive mechanisms. After reading about persuasive interfaces, behavior models and behavior change techniques, the *PIQL* system was implemented. The system consists in the creation of persuasive messages to encourage the user to improve his behavior or keep it if it's positive already. The implementation carefully follow *PersonAAL*'s goal. Looking back at the work already made and looking at the future work for this system it can be considered that *PIQL*'s goals were successfully achieved, currently existing a persuasive system integrated with *PersonAAL* validated according to different kinds of tests, being the overall evaluation very satisfying. While the results are satisfying, *PIQL* can possess new functionalities, which can improve the system.

Keywords: *PIQL*, Persuasive Interfaces, Rule System, Persuasion, Quality of Life.

Conteúdo

Capítulo 1	Introdução	1
1.1	Objetivos.....	4
1.2	Estrutura do Documento	5
Capítulo 2	Trabalho Relacionado	7
2.1	Modelos de Comportamento	7
2.1.1	Modelo Contínuo.....	7
2.1.2	Modelo por etapas	8
2.1.3	HAPA (<i>Health Action Process Approach</i>).....	10
2.1.4	COM-B.....	11
2.1.5	Modelo de Crença de Saúde.....	12
2.1.6	Modelo Comportamental de <i>Fogg</i>	13
2.2	Técnicas de Alteração de Comportamentos	15
2.3	Interfaces Persuasivas.....	17
Capítulo 3	<i>PersonAAL</i>	21
Capítulo 4	Desenho.....	25
4.1	Conceptualização.....	25
4.2	Módulos do <i>PIQL</i>	26
4.2.1	Identificação de Padrões.....	27
4.2.2	Deteção de desvios	30
4.2.3	Modelação de comportamento	40
4.2.4	Seleção de técnicas de alteração de comportamentos	41
4.2.5	Implementação de técnicas de alteração de comportamento	42
Capítulo 5	Implementação	43
5.1	Ferramentas utilizadas	43
5.2	Módulos	45
5.3	Base de dados	55
5.4	Interface <i>REST</i>	58

5.5	Servidor	63
5.5.1	Ficheiros inseridos no servidor	63
5.6	Execução do sistema.....	65
Capítulo 6	Avaliação.....	71
Capítulo 7	Conclusão.....	75
7.1	Trabalho Futuro	77
Bibliografia	79
Apêndice	81
Apêndice A	– Ficheiro Excel de Atividade Física.....	81
Apêndice B	– Lista de Fatores que afetam comportamento	82
Apêndice C	– Lista de Técnicas de Persuasão	86
Apêndice D	– Criação de cápsulas.....	86
Apêndice E	– Lista de Regras e Fatores – 1ª Reunião	88
Apêndice F	– Lista de Regras e Fatores – 2ª Reunião	91

Lista de Figuras

Figura 1. Esquema representativo da Teoria do Comportamento Planeado ⁸	8
Figura 2. Imagem ilustrativa do Modelo HAPA (Farinha, 2012).....	10
Figura 3. Imagem Representativa do modelo COM-B	11
Figura 4. Figura representativa do Modelo de Crença de Saúde	12
Figura 5. Imagem representativa do Modelo Comportamental de Fogg (Fogg, 2009)	13
Figura 6. Imagem ilustrativa do MoviPill (de Oliveira et al., 2010)	17
Figura 7. Imagem ilustrativa dos mecanismos do Waterbot (Visível mecanismo em que iluminam água de vermelho caso não esteja a ser usada corretamente). (Arroyo et al., 2005).....	18
Figura 8. Exemplo da aplicação ViTo. Lado Esquerdo - Menu Inicial; Lado Direito - as opções que o utilizador possui. (Nawyn et al., 2006)	19
Figura 9. Plataforma PersonAAL. Vermelho – Localização do Sistema PIQL, que funcionará como Modulo de Análise de Comportamento e de Persuasão	21
Figura 10. Esquema representativo do conceito inicial do sistema PIQL.	26
Figura 11. Esquema representativo da conversão do conceito inicial (Figura 10) para os módulos atuais do PIQL.....	27
Figura 12. Dendograma que mostra os dois distintos grupos de atividade baseados nos últimos 28 dias de atividade do utilizador.	29
Figura 13. Comparação entre dois dias (Dia A a azul e B a vermelho). O Dia B (a vermelho) foi adiantado 2 horas (representado a vermelho tracejado) para realizar uma comparação mais semelhante entre os dois dias.....	30
Figura 14. Um dos casos de teste avaliado. A vermelho – Dia representativo do cluster. A azul – Novo dia. Pela inconsistência da atividade e desvio comparativo ao dia representativo, este dia é classificado de desvio grave.....	39
Figura 15. Aspeto do projeto gradle acerca do sistema PIQL no ambiente de desenvolvimento integrado Eclipse	44
Figura 16. Representação gráfica da aproximação 1 (feita através da plataforma R).	47

Figura 17. Imagem representativa da aproximação 2. Preto – Energia do dia representativo; Azul – Energia do Novo dia; Amarelo – Limite do desvio pequeno; Vermelho – Limite do desvio grande. Fornecido através da Plataforma R.....	48
Figura 18. Imagem representativa da aproximação 3. Preto – Passos do dia representativo; Azul – Passos do Novo dia; Vermelho Superior – Limite superior; Vermelho Inferior – Limite Inferior. Fornecido através da Plataforma R.....	49
Figura 19. Exemplo de um ficheiro JSON com a árvore de decisão a ser analisada.	51
Figura 20. Ficheiro de Configuração do módulo DeviationPatternFinals. Possível observar as constantes para cada aproximação, juntamente com a aproximação a ser utilizada	52
Figura 21. Lista de regras que explicam o possível desvio do utilizador.	53
Figura 22. Regra LocalWeatherRule1. Retorna verdadeiro caso o tempo esteja bom e os passos dados estejam abaixo do objetivo. Caso contrário retorna falso. O método booleano deve possuir o mesmo nome que a regra e o caracter inicial minúsculo (localWeatherRule1).....	54
Figura 23. Tabela com as mensagens associadas a cada fator e técnica persuasiva.	55
Figura 24. Esquema fornecido pelo Adminer relativo à base de dados. A verde estão variáveis do tipo “text”, a roxo “data”, as restantes são inteiros ou strings	56
Figura 25. Esquema representativo do sistema PIQL através da interface REST. A numeração representa os passos de execução do sistema.....	59
Figura 26. SourceTree, que possui o repositório que faz a transferência de ficheiros do ambiente de desenvolvimento para o servidor do PIQL.....	63
Figura 27. Visualização dos ficheiros necessários para a execução do sistema PIQL	63
Figura 28. Visualização do comando de exemplo 1, desenhado para calcular o padrão de atividade dos utilizadores na base de dados. Existe 3 outputs dado que a base de dados possui 3 utilizadores. Após execução, devido à opção Module, o programa sai.	67
Figura 29. Execução do 2º comando de exemplo. Nesta parte é possível ver a análise dos fatores que podem explicar o desvio de comportamento do Terrence (utilizador no ficheiro de configuração).....	67

Figura 30. Continuação do output obtido da execução do 2º comando do exemplo. Possível observar as regras que possuem parâmetros válidos, juntamente com as que foram despoletadas, BCTs que as regras retornam e aviso de mensagem persuasiva.... 68

Figura 31. Parte final do output obtido da execução do 2º comando do exemplo. Possível observar aspetos da figura anterior. Possível observar a mensagem persuasiva obtida e a aceitação da mensagem por parte do Adaptation Engine. 68

Figura 32. Visualização dos comandos de crono realizados no servidor 69

Lista de Tabelas

Tabela 1 – Fatores que influenciam a saúde nos idosos de acordo com a OMS ¹	2
Tabela 2 – Necessidades para o envelhecimento saudável ¹	3
Tabela 3 – Tabela Representativa do Modelo Transteórico. Possui caso exemplificativo de um indivíduo que quer deixar de fumar.....	9
Tabela 4 – Modelos de Comportamento Estudados.....	14
Tabela 5 – Aspectos a serem realizados no projeto juntamente com o modelo que suporta esse aspeto.	14
Tabela 6 – Técnicas de Alteração de Comportamentos	16
Tabela 7 – Resumo das aproximações usadas para detetar o desvio no Módulo DeviationPatternsFinal	32
Tabela 8 – Taxa de Sucesso em cada aproximação	40
Tabela 9 - Cápsulas do sistema e seus objetivos.....	64
Tabela 10 – Opções de execução do sistema PIQL	65
Tabela 11 – Argumentos para a execução da opção CommandLine	66
Tabela 12 - Perguntas realizadas no inquérito a idosos.	73
Tabela 13 – Respostas à pergunta “Gostaria de receber este tipo de mensagens”. 73	
Tabela 14 - Descrição dos aspetos a serem realizados no projeto PIQL e onde estes estão incluídos no projeto	77

Capítulo 1

Introdução

Um estudo da Organização Mundial de Saúde (OMS) estima que entre 2000 e 2050, o número de pessoas com mais de 60 anos cresça para o dobro e uma em cada quatro pessoas no mundo terá mais que 60 anos¹. Em Portugal a previsão não é diferente, onde o Instituto Nacional de Estatística (INE) acredita que para 2080, a população portuguesa diminua (de 10,3 para 7,5 milhões de pessoas) porém a população idosa aumente consideravelmente (de 2,1 para 2,8 milhões), onde por cada 100 jovens, teremos 317 idosos².

Este crescimento não se regista apenas ao nível do aumento de população idosa, mas também ao nível da esperança média de vida, tendo a OMS confirmado que entre 2000 a 2015 a esperança média de vida mundial aumentou 5 anos³ e o INE acredita que o mesmo se registre a nível nacional, prevendo que em 2080² esta se situe nos 92 anos para as mulheres e 87 anos para os homens.

Em contraste, a qualidade de vida dos idosos não está a aumentar ao mesmo ritmo que os crescimentos anteriormente referenciados. No relatório mundial de Envelhecimento e Saúde de 2015 (*World Report on Ageing and Health 2015*) da OMS, contra as expectativas, o relatório demonstrou que existem poucas provas em como a quantidade extra de anos que um indivíduo está a usufruir, são acompanhados por um

¹ Informação encontrada no infográfico “healthy ageing infographic” da Organização Mundial de Saúde (OMS), visível em <http://www.who.int/ageing/events/world-report-2015-launch/healthy-ageing-infographic.jpg?ua=1>, visto em 29 de Julho de 2017.

² Dados estatísticos do INE, visíveis em https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_destaques&DESTAQUESdest_boui=277695619&DESTAQUESmodo=2 e utilizados também pelo “Público” <https://www.publico.pt/2017/03/29/sociedade/noticia/portugal-estara-reduzido-a-75-milhoes-de-pessoas-em-2080-1766915>. Visto em 29 de Julho de 2017

³ Notícia da OMS sobre esperança média de vida e saúde associada, visível em <http://www.who.int/mediacentre/news/releases/2016/health-inequalities-persist/en/>, visto em 29 de Julho de 2017.

aumento da qualidade dos mesmos⁴. Esta ideia é reforçada por John Beard, Diretor do Departamento de Envelhecimento e de Vida da OMS, que afirmou *“Infelizmente 70 não parece ser os novos 60. Mas podia ser. E devia ser”*⁴. Este pensamento é corroborado através de dados, onde em países como os Estados Unidos da América, é estimado que em 2030, 80% dos idosos terão uma condição crónica e que 50% da população idosa terá duas ou mais (Lee, Helal, Anton, De Deugd, & Smith, 2012), afetando diretamente idosos, criando incapacidade, diminuição da qualidade de vida e aumento dos custos dos serviços de saúde (Lee et al., 2012).

Embora estes sejam os resultados atuais, a antiga Diretora-Geral de Saúde da OMS, Doutora Margaret Chan, afirmava *“Presentemente, a maioria das pessoas, mesmo nos países mais pobres, estão a viver vidas mais longas. Mas isto não é suficiente. Precisamos de assegurar que estes anos extras são saudáveis, com significado e dignificantes. Ao assegurarmos isto, não será só bom para os idosos, mas para a sociedade no geral”*⁴. Esta forma de pensar levou a que a OMS indicasse fatores que influenciam a saúde em idosos (evidenciados na Tabela 1), bem como os aspetos necessários para que haja um envelhecimento com qualidade (referidos na Tabela 2).

Fatores que influenciam a saúde nos idosos			
Individual		Ambiente em que vivem	
Mudanças relacionadas com a idade	Genética	Habitação	Tecnologias de assistência
Comportamentos	Doenças	Transportes	Estruturas Sociais

Tabela 1 – Fatores que influenciam a saúde nos idosos de acordo com a OMS¹

⁴ Aumento da esperança média de vida e fatores para aumentar a qualidade de vida, visível em <http://www.who.int/mediacentre/news/releases/2015/older-persons-day/en/>, visto em 29 de Julho de 2017.

<p>Necessidades para o envelhecimento saudável</p>	Mudança na forma como olhamos para o envelhecimento e para os idosos
	Criação de ambientes amigáveis ao envelhecimento
	Convergência dos sistemas de saúde para as necessidades dos idosos
	Desenvolvimento de sistemas para tratamentos a longo-prazo

Tabela 2 – *Necessidades para o envelhecimento saudável¹*

É no contexto de todos os aspetos acima supracitados que foi criado o projeto *PersonAAL*, projeto esse que está diretamente envolvido com o trabalho realizado nesta tese. O *PersonAAL* ⁵ é um projeto que visa estender o tempo que os idosos conseguem viver no ambiente da sua casa, aumentando a sua autonomia e assistindo-os nas atividades do dia-a-dia. O projeto permite a ligação com aplicações web inteligentes, fornecendo a estas dados personalizados e do contexto do idoso, permitindo às aplicações fornecer informação ao utilizador na sua própria casa, com o objetivo de melhorar a qualidade de vida, a autonomia e diminuir os custos dos serviços de saúde associados (aspeto importante, uma vez que além de se prever que em 2050, 80% dos idosos vão viver em países com baixo-médio subsídio¹, os governos pretendem cortar ao máximo os custos associados a esse tipo de serviços⁶ sem perder a sua qualidade). Para atingir os objetivos acima supracitados, o *PersonAAL* está a desenvolver uma plataforma que irá permitir a aplicações orientadas a idosos, serem suportadas por mecanismos de persuasão e adaptação. Por exemplo uma aplicação web orientada a idosos, que se foque nos aspetos de reabilitação, ao aceder à plataforma *PersonAAL* seria capaz de mandar avisos personalizados de acordo com as necessidades do utilizador, adaptar a interface do utilizador consoante as qualidades, deficiências do idoso ou meio que o rodeia.

Como evidenciados na Tabela 1, comportamentos e tecnologias de assistência são fatores capazes de influenciar a saúde dos idosos. Há provas que as tecnologias persuasivas e o uso associado a um sistema de computador, aparelho ou aplicação desenhada para persuadir o comportamento de uma pessoa, possuem potencial para mudar o comportamento dos seus utilizadores (Chatterjee & Price, 2009). Muitos idosos preferem viver independentes de encarregados de saúde, ao ponto da indústria focalizar

⁵ Informações adicionais do projeto em www.personaal-project.eu/

⁶ Exemplo de cortes em serviços de saúde em <https://www.nytimes.com/2017/06/24/science/medicaid-cutbacks-elderly-nursing-homes.html> , visto em 30 de Julho de 2017.

esforços na procura de formas de suporte de vida independente, bem como fornecer soluções de baixo custo para a promoção do envelhecimento saudável (Lee et al., 2012). O projeto *PersonAAL* está ciente da importância destes fatores, ao ponto da plataforma vir a ter, entre outros, um módulo de persuasão, onde se espera identificar situações em que mensagens persuasivas devem ser dadas aos utilizadores, de modo a mudarem o seu comportamento atual caso este seja negativo ou reforçá-lo em caso positivo. É aqui que o projeto *PIQL* entra. O projeto *PIQL* visa a exploração e desenho de interfaces persuasivas, com o intuito de incentivar os utilizadores a adotarem estilos de vida saudáveis, quer do ponto de vista físico como do ponto de vista cognitivo. A persuasão é feita através de um sistema que terá como objetivo final a criação de mensagens motivacionais geradas a partir de regras e do contexto do utilizador, tendo como público-alvo a população idosa.

1.1 Objetivos

Por tudo o que foi atrás mencionado o projeto *PIQL* tem como objetivos:

- Identificar interfaces persuasivas existentes que se possam adequar ao contexto do projeto *PersonAAL*.
- Criação de um módulo/sistema persuasivo através da implementação ou alteração de interfaces persuasivas que se adequem ao contexto do projeto.
- Implementar esse módulo/sistema no projeto *PersonAAL*.

O cumprimento dos objetivos anteriores põem o sistema *PIQL* numa posição favorável para integrar a plataforma *PersonAAL*.

1.2 Estrutura do Documento

Neste documento estão incluídos capítulos fundamentais para a compreensão do projeto *PIQL*. No Capítulo 2 (Trabalho Relacionado) é descrita a informação recolhida acerca de tópicos importantes para o desenvolvimento do projeto, tais como Interfaces Persuasivas, Modelos de Comportamento e Técnicas de Alteração de Comportamentos. No Capítulo 3 (*PersonAAL*) define-se o projeto *PersonAAL*, juntamente com a arquitetura do projeto e módulos associados, e a relação deste com o *PIQL*. O Capítulo 4 (Desenho) contém informação acerca da arquitetura do sistema e aspetos particulares do sistema. O Capítulo 5 (Implementação) possui as escolhas relacionadas com a implementação do sistema, tais como linguagens utilizadas, *frameworks*, restrições e outros aspetos. Capítulo 6 (Resultados) reporta os vários testes realizados ao sistema, bem como descrição dos resultados e problemas que surgiram. O último Capítulo (Conclusão) apresenta a discussão dos resultados, o cumprimento dos objetivos delineados, bem como mudanças que devem ser tomadas e aspetos a serem trabalhados no futuro.

Capítulo 2

Trabalho Relacionado

Este capítulo centra-se na pesquisa acerca de áreas úteis ao desenvolvimento e cumprimento dos objetivos delineados. As áreas consideradas interessantes são as áreas dos modelos de comportamento, técnicas de alteração de comportamentos e interfaces persuasivas.

2.1 Modelos de Comportamento

Esta área é importante no desenvolvimento do projeto, pois é necessário compreender os fatores e etapas que levam um indivíduo a tomar uma certa decisão. O comportamento pode ser definido como “*reação de um indivíduo ou animal perante uma situação ou conjunto de estímulos*”⁷. Pela definição, podemos partir do princípio que o comportamento parte da reação a um conjunto de estímulos. Devido ao elevado número de estímulos houve a necessidade de criar modelos de comportamentos. O estudo realizado à volta destes modelos revelou que estes, para além de numerosos (Farinha, 2012; Hkdylru, 2006; Morris, Marzano, Dandy, & O’Brien, 2012; S.Sutton, 2002; Schwarzer, 2008), são usados nas mais variadas áreas. Atendendo aos objetivos delineados na introdução, as áreas em que nos interessa analisar modelos de comportamentos são a área da saúde, mudança de comportamentos e persuasão. Dentro destas e derivado do enorme número de modelos, analisámos seis modelos: Modelo Contínuo, Modelo Por Etapas, HAPA, COM-B, Modelo de Crença de Saúde e Modelo Comportamental de Fogg.

2.1.1 Modelo Contínuo

O Modelo Contínuo é constituído por um conjunto de teorias, as quais sugerem que o comportamento ou ação de uma pessoa é relacionado com intenções conscientes que a pessoa tem (Schwarzer, 2008). Estas intenções são derivadas de crenças e atitudes tais como barreiras, normas sociais, gravidade de doenças, vulnerabilidades pessoais ou auto

⁷ Definição do Dicionário, versão psicologia. Visível em <http://www.infopedia.pt/dicionarios/lingua-portuguesa/comportamento>. Visto em 31 de Julho de 2017.

eficácia. Essas intenções são combinadas numa equação de palpite para explicar intenções e mudanças comportamentais (Schwarzer, 2008). Dentro do conjunto de teorias, as mais destacadas são a Teoria da Ação Racionalizada, Teoria do Comportamento Planeado e a Teoria da Motivação para Proteção. A mais conhecida deste conjunto de teorias, a Teoria do Comportamento Planeado, afirma que o comportamento é determinado pela intenção do agente em realizar o comportamento (sendo a intenção o antecessor imediato do comportamento). Essa intenção, como é possível observar na Figura 1, é determinada por três fatores: a atitude do agente acerca do comportamento, a norma subjetiva (pressão que tem a realizar o comportamento) e o seu controle acerca do comportamento (percepção que a pessoa tem da sua habilidade em realizar o seu comportamento). A intenção da pessoa em realizar o comportamento que deseja, e o mesmo ser realizado, está diretamente relacionado com o quão favorável os fatores anteriormente mencionados são⁸.

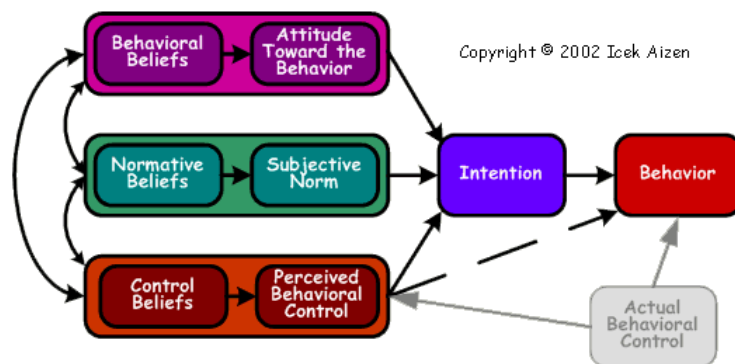


Figura 1. Esquema representativo da Teoria do Comportamento Planeado⁸

O aspecto a reter neste modelo para o projeto é que através de um conjunto reduzido de variáveis baseadas em antecedentes, podemos ter uma previsão das intenções comportamentais, mas não devemos possuir um modelo que acredite que um indivíduo não pode mudar o seu comportamento de forma radical, pois estamos a limitar a persuasão que certos fatores podem ter num indivíduo. Por exemplo nega que o utilizador pode deixar de fumar ou beber repentinamente.

2.1.2 Modelo por etapas

O Modelo por Etapas é referente a teorias que assentam na ideia de que quando há uma mudança de comportamento, esta passa por etapas, sendo que só se passa de uma etapa para a próxima através de processos cognitivos e experienciais (Schwarzer, 2008).

⁸ Universidade de Twente sobre essa teoria. https://www.utwente.nl/en/bms/communication-theories/sorted-by-cluster/Health%20Communication/theory_planned_behavior/. Visto em 31 de Julho de 2017.

O mais conhecido destes modelos é o Modelo Transteórico. Ele afirma que existem 5 etapas: Pré-Contemplanção, Contemplanção, Preparação, Ação, Manutenção (como evidenciadas na Tabela 3) (Schwarzer, 2008).

Etapas do Modelo	Explicação	Exemplo de Comportamento
Pré-Contemplanção	Indivíduo não entende o problema. Não possui intenção de mudar comportamento num futuro próximo.	Não pensa sequer em parar de fumar.
Contemplanção	Indivíduo entende o problema. Séria consideração em mudar o comportamento.	Começa a examinar o uso do tabaco e o desejo de sair. Pesa os prós e contras de deixar de fumar
Preparação	O indivíduo pretende tomar ações.	Fez o compromisso de deixar de fumar e começa a desenvolver um plano para deixar de fumar.
Ação	Indivíduos modificam os seus comportamentos, experiências ou ambiente de forma a superar os problemas.	Põe o plano de deixar de fumar em ação.
Manutenção	Trabalho realizado para prevenir recaídas e consolidar os ganhos.	Manteve-se abstinente após deixar de fumar.

Tabela 3 – Tabela Representativa do Modelo Transteórico. Possui caso exemplificativo de um indivíduo que quer deixar de fumar.

Deste modelo retiramos que indivíduos na mesma etapa, possuem as mesmas barreiras, logo podem ser ajudados da mesma forma, porém salienta-se o facto de, como no modelo anterior, não permitir mudanças bruscas de comportamento (neste caso salto entre etapas ou recessão) e não contempla fatores externos que possam mudar o comportamento (económicos, sociais, ambientais).

2.1.3 HAPA (*Health Action Process Approach*)

Este modelo sociocognitivo (perceptível na Figura 2) sugere que mudanças de comportamento de saúde são um processo que consiste numa fase motivacional e numa fase volitiva (de vontade) (Farinha, 2012). Este modelo faz a ponte entre o Modelo Contínuo e o Modelo por Etapas, sendo mais focado no modelo por etapas.

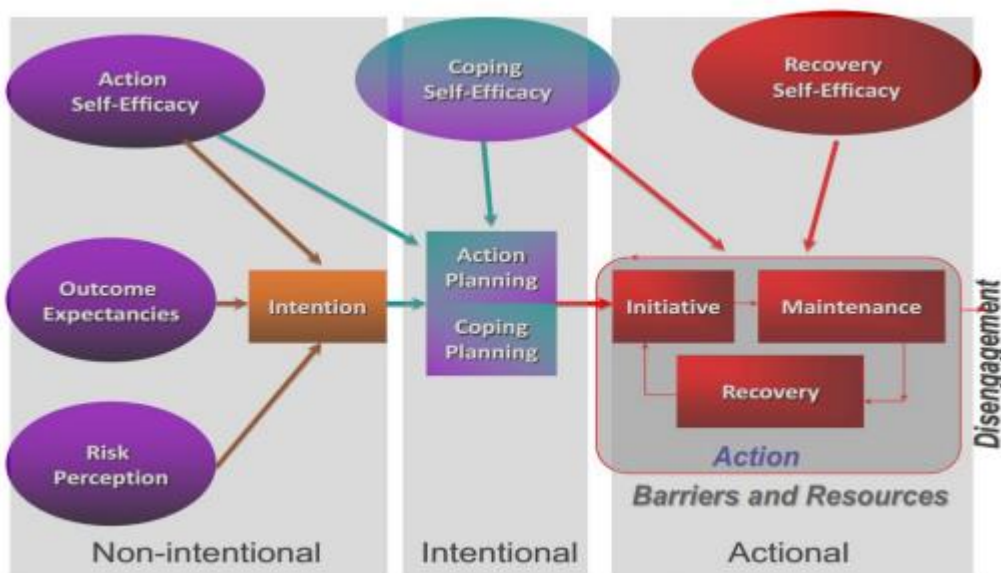


Figura 2. Imagem ilustrativa do Modelo HAPA (Farinha, 2012)

Na fase motivacional, caracterizada por ser uma fase que antecede a intenção, o indivíduo deverá desenvolver a intenção de executar um comportamento. O estado correspondente a esta fase é portanto denominado de pré-intencional (na Figura 2 possui o nome não-intencional). A fase seguinte é composta por dois estados. No estado intencional o indivíduo elabora estratégias à iniciação do comportamento. No estado de ação o indivíduo utiliza estratégias para a manutenção do comportamento já iniciado (Farinha, 2012). Esta fase tem como intuito prevenir recaídas e prover o indivíduo de estratégias e mecanismos de autorregulação: planeamento de estratégias, tomar iniciativa, ultrapassar obstáculos, monitorizar os passos iniciais, prevenir e recuperar de eventuais recaídas. A passagem da fase Intencional para a fase de Ação ocorre através da iniciação do comportamento desejado.

Deste modelo podemos retirar que há fases que podem prever comportamentos (como referido no Modelo Contínuo). Também retiramos que parte do utilizador mudar o seu tipo de comportamento, e caso haja essa mudança, podemos dar-lhe as ferramentas necessárias para manter esse comportamento.

2.1.4 COM-B

O modelo (observável na Figura 3) reconhece que o comportamento é parte de um sistema interativo que envolve as componentes da capacidade (psicológica e física), oportunidade (física ou social) e a motivação (que pode ser automática ou refletiva)⁹. Afirma que o comportamento provém da capacidade de realizar um comportamento, bem como a oportunidade e a motivação de executar o comportamento (UCL Working Group, 2015). Novos comportamentos ou mudanças de comportamento requerem uma mudança numa ou mais das componentes anteriormente referidas (UCL Working Group, 2015). O modelo pode ser usado para identificar objetivos (qual componente é o melhor alvo para mudança) e que tipos de intervenção podem ser mais eficientes, bem como a duração da implementação da intervenção. Ele é especialmente útil na identificação de objetivos e decisão de intervenções (que capacidades, oportunidades e motivação são mais relevantes para o alvo da mudança comportamental e quais destes podem apresentar um maior obstáculo à mudança) (UCL Working Group, 2015).

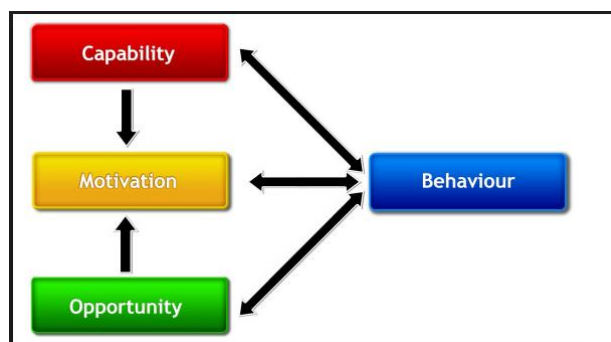


Figura 3. Imagem Representativa do modelo COM-B¹⁰

Para o projeto este modelo é fundamental em vários aspetos. Dos modelos estudados, este é o primeiro modelo que reconhece o comportamento como um sistema interativo, onde há uma relação interativa entre fatores de comportamento e utilizador. É de fácil compreensão a relação entre cada fator e como a falta de um desses fatores explica determinado comportamento. Outro aspeto importante é o modelo permitir identificar objetivos, o que dá espaço para personalizar o sistema para cada utilizador, bem como reconhecimento dos tipos de intervenções que devem ser feitas.

⁹ Retirado do site de compra do livro referente ao COM-B, visível em <http://www.behaviourchangewheel.com/about-wheel>. Visto em 31 de Julho de 2017.

¹⁰ Imagem representativa do modelo COM-B, visível em https://www.researchgate.net/figure/51070630_fig1_The-COM-B-system-a-framework-for-understanding-behaviour. Visto em 3 de Setembro de 2017.

2.1.5 Modelo de Crença de Saúde

O Modelo de Crença de Saúde (ilustrado na Figura 4) é um modelo psicológico que tenta explicar e prever comportamentos de saúde, sendo este feito através do foco nas atitudes e crenças dos indivíduos ¹¹. O modelo é baseado na compreensão que a pessoa vai tomar uma ação relacionada com a sua saúde se a pessoa:

- Sente que uma condição de saúde negativa pode ser evitada.
- Tem uma expectativa positiva que se tomar uma ação recomendada, ele/ela vai evitar uma condição de saúde negativa.
- Acredita que pode com sucesso fazer a ação de saúde recomendada.

A operacionalização deste modelo baseia-se na realização de inquéritos e a partir daí ver os fatores que modificam comportamentos e a probabilidade de uma ação ocorrer.

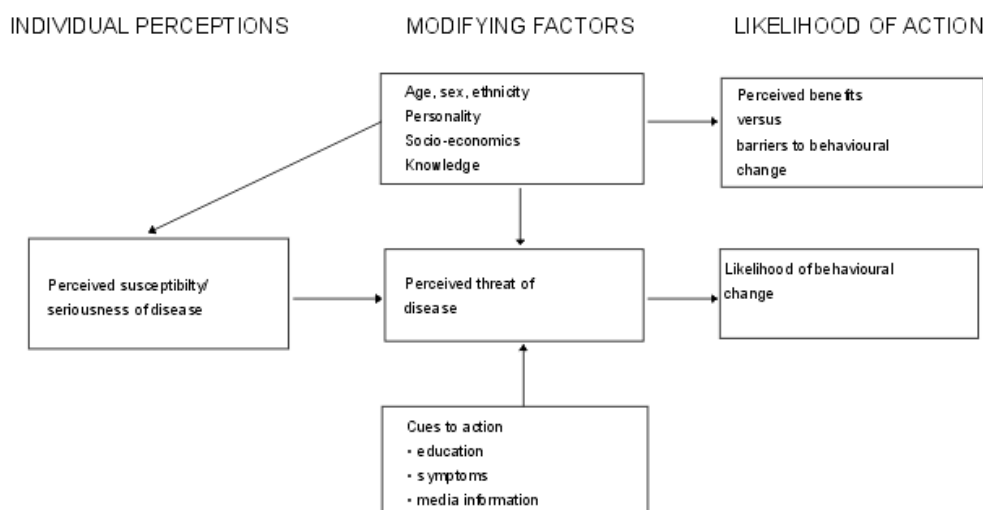


Figura 4. Figura representativa do Modelo de Crença de Saúde

Este modelo é importante pois algumas das suas desvantagens são aspetos a ter em consideração no projeto. Este modelo não tem em conta como emoções e ansiedade afetam o pensamento racional e a tomada de decisão do indivíduo¹², bem como a forma de realização deste modelo submete o utilizador a processos repetitivos (sujeita-o a vários inquéritos).

¹¹ Informações do Modelo de Crença de Saúde retiradas de https://www.utwente.nl/en/bms/communication-theories/sorted-by-cluster/Health%20Communication/Health_Belief_Model/. Visto em 31 de Julho de 2017

¹² Referente ao Modelo de Crença de Saúde, retirado de <http://amactraining.co.uk/resources/handy-information/free-learning-material/models-and-theories-of-health-behaviour-change-index/models-and-theories-of-health-behaviour-2/>. Consultado em 31 de Julho de 2017.

2.1.6 Modelo Comportamental de *Fogg*

Este modelo psicológico (visível na Figura 5) refere que o comportamento é derivado de três fatores: motivação, capacidade e o estímulo (conhecido como *Trigger*) (Fogg, 2009). Cada um destes é composto por subcomponentes. A motivação está relacionada com a pré-disposição que a pessoa tem para realizar o comportamento. Por mínima que seja, a pessoa deve ter algum tipo de motivação para realizar o comportamento (Fogg, 2009). É habitual achar-se que a motivação gera o comportamento, mas a capacidade é a mais importante (Fogg, 2009). A capacidade está relacionada com a facilidade que um indivíduo possui para realizar um comportamento. A relação entre capacidade e motivação existe. Podemos fazer comportamentos com baixo nível de motivação, mas com alta capacidade e vice-versa. O comportamento fica mais fácil de ser realizado caso haja altos níveis de motivação e capacidade (Fogg, 2009). Finalmente temos o estímulo. Este consiste numa espécie de aviso que diz às pessoas para realizarem um certo comportamento. Os estímulos têm em comum três características: apercebermo-nos deles, associarmos-los a um comportamento e este aparecer quando estamos motivados e prontos para realizarmos o comportamento. A importância está no *timing* do estímulo, sendo esta a componente fundamental do estímulo.

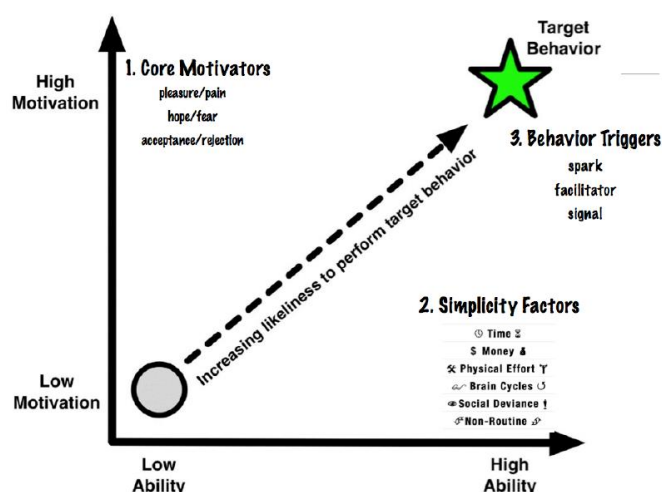


Figura 5. Imagem representativa do Modelo Comportamental de Fogg (Fogg, 2009)

O ponto a retirar é que com a conjugação certa de fatores, podemos fazer com que o utilizador fique tentado a corrigir o seu comportamento.

Em suma através do estudo de modelos de comportamento (visíveis na Tabela 4) houve um desenvolvimento no conhecimento acerca dos fatores que influenciam o comportamento do indivíduo bem como as vantagens e desvantagens que alguns modelos de comportamento possuem. Foi possível com este estudo retirar aspetos a serem realizados no projeto (enumerados na Tabela 5).

Modelos de Comportamento Estudados	
Modelo Contínuo	Modelo Por Etapas
HAPA	COM-B
Modelo de Crença de Saúde	Modelo Comportamental de <i>Fogg</i>

Tabela 4 – Modelos de Comportamento Estudados

Aspetos a serem realizados no projeto	
Aspetos Retirados	Fonte
Através de variáveis antecedentes, prever intenções comportamentais	2.1.1 Modelo Contínuo 2.1.3 HAPA (<i>Health Action Process Approach</i>)
Indivíduos na mesma etapa do comportamento podem ser ajudados da mesma forma	2.1.2 Modelo por etapas
Contemplar fatores que possam mudar o comportamento (tanto pessoais como exteriores)	2.1.2 Modelo por etapas 2.1.4 COM-B 2.1.5 Modelo de Crença de Saúde
Dar as ferramentas ao utilizador para manter o seu comportamento correto	2.1.3 HAPA (<i>Health Action Process Approach</i>)
Não possuir um modelo comportamental que acredite que um indivíduo não pode mudar o seu comportamento de forma radical	2.1.1 Modelo Contínuo 2.1.2 Modelo por etapas
Possuir um modelo comportamental fácil de programar, dinâmico, que identifique objetivos e seja personalizado para cada utilizador.	2.1.4 COM-B
Reconhecer tipos de intervenções que devem ser feitas ao utilizador	2.1.4 COM-B
Conjugar fatores para tentar apelar a mudança de comportamentos errados	2.1.4 COM-B 2.1.6 Modelo Comportamental de <i>Fogg</i>

Tabela 5 – Aspetos a serem realizados no projeto juntamente com o modelo que suporta esse aspeto.

2.2 Técnicas de Alteração de Comportamentos

Técnicas de alteração de comportamento, também conhecidas como técnicas persuasivas, são técnicas gerais ou processos designados a influenciar as variáveis de um comportamento, de modo a incentivar um comportamento diferente do que o utilizador normalmente apresenta (Kok, 2014). Em termos tecnológicos, tecnologia de persuasão computacional é um sistema computacional, aparelho ou aplicação intencionalmente desenhada para mudar a atitude/comportamento de uma pessoa de uma determinada forma (Chatterjee & Price, 2009). Existem muitas técnicas de alteração de comportamentos (Arroyo, Bonanni, & Selker, 2005; de Oliveira, Cherubini, & Oliver, 2010; Kok, 2014; Michie et al., 2013; Schwarzer, 2008; Swamy Ananthanarayan & Siek, 2012). A maioria das teorias sociocognitivas dizem que a intenção do indivíduo em querer mudar é a melhor direção para que haja uma mudança de comportamentos, embora o comportamento das pessoas por vezes não vai ao encontro das suas intenções (devido a barreiras ou influências) (Sutton, 2002). Tendo como base a secção anterior, podemos observar a associação entre comportamento e utilizador, pelo que faz todo o sentido que as técnicas de alteração de comportamentos centradas no utilizador, e nas suas intenções, sejam as mais importantes, e sobre as quais esta revisão de literatura se irá focar.

Uma das técnicas referidas é o Aumento de Consciencialização (Kok, 2014; Swamy Ananthanarayan & Siek, 2012). Consiste em fornecer informação, *feedback* ou confrontação acerca das causas, consequências e alternativas de um problema ou comportamento problemático (Kok, 2014). Os parâmetros usados são o *feedback*, confrontação, aumento de consciência seguido do aumento da habilidade de resolver ou problema bem como autoeficácia.

Outra técnica é o Reforço Positivo (Arroyo et al., 2005). Esta consiste em, durante a intenção de realizar uma atividade ou durante a mesma, haver algo que o utilizador deseje (recompensa pelo comportamento feito por exemplo). Se presente no tempo que ocorreu a ação, o reforço positivo tende a aumentar a probabilidade da ação ser repetida. Esta técnica é associada por exemplo a recompensas, como por exemplo recompensa material, social, não-definida ou Auto recompensa (Michie et al., 2013).

Em contraste ao reforço positivo, temos o Reforço Negativo (Arroyo et al., 2005). O reforço negativo é criado aquando da execução de comportamentos mais negativos. Se durante o comportamento realizado o indivíduo receber indicações negativas acerca do comportamento, pode ter um efeito “negativo” e fazer com que o utilizador deixe de recriar esse comportamento. Ameaça é a mais utilizada, que passa por mostrar ao utilizador as consequências que um certo comportamento pode trazer-lhe.

Outra técnica é a Validação Social (Arroyo et al., 2005). Esta técnica passa um pouco pela influência social, ou seja, o comportamento é dito correto se as outras pessoas também considerariam correto. É importante em casos em que a cultura onde o indivíduo está inserido adota comportamentos corretos.

Feedback e Monitorização (Michie et al., 2013) é também uma técnica de alteração de comportamentos, em que o objetivo é monitorizar o indivíduo e tentar em tempo real fornecer os dados das suas atividades a si mesmo. Esta técnica pode possuir varias formas tais como *feedback* comportamental, bio *feedback* e auto monitorização do comportamento (Michie et al., 2013).

Objetivos e Planeamento são consideradas técnicas. Planeamento de objetivos (resultados e comportamento), planeamento de ações, compromisso e diferença entre comportamento atual e comportamento pretendido são técnicas que podem delinear comportamentos de saúde normais para o utilizador e cumprimento destes (Michie et al., 2013).

Identidade e Auto Crença são também técnicas de forma a alterar comportamentos. Planeamento mental de sucesso, auto conversas, persuasão verbal para aumentar auto-eficácia e auto-confiança são todas elas técnicas que tentam aumentar a confiança do indivíduo de modo a persuadí-lo que consegue realizar os comportamentos ao qual se propôs (Michie et al., 2013).

Existem mais tipos de técnicas que não foram mencionadas devido ao elevado número de técnicas, sendo as técnicas apresentadas neste subcapítulo (e observáveis na Tabela 6) consideradas as mais importantes tendo em conta a previsão de eficácia que estas técnicas teriam no âmbito do projeto *PIQL*.

Técnicas de Alteração de Comportamentos	
Aumento de Consciencialização	<i>Feedback</i> e Monitorização
Reforço Positivo	Objetivos e Planeamento
Reforço Negativo	Identidade e Auto-Crença
Validação Social	

Tabela 6 – Técnicas de Alteração de Comportamentos

2.3 Interfaces Persuasivas

Como referido na introdução, o *PIQL* visa a exploração e desenho de interfaces persuasivas. O *PIQL* suporta a criação de mensagens persuasivas, tornando importante a recolha de informação sobre aspetos persuasivos inseridos em interfaces, mas também de aspetos que não comprometam a persuasão de interfaces. Neste capítulo irá ser abordado a ligação entre comportamento, persuasão e interface.

O primeiro caso estudado foi o *MoviPill* (de Oliveira et al., 2010). O jogo (visível na Figura 6) baseava-se em competência social, onde a aposta numa forma mais divertida em vez do tradicional aviso de toma de medicação e a ideia de competição, captou a atenção do utilizador. Eventualmente é o reforço positivo (diferentes recompensas para diferentes pessoas) bem como a ideia de socialização (troca de informações entre utilizadores) que poderiam levar o utilizador a mudar o seu comportamento, juntamente com os fatores que persuadiram o utilizador a usar a aplicação. A simplicidade da interface foi também um dos fatores do sucesso desta interface.

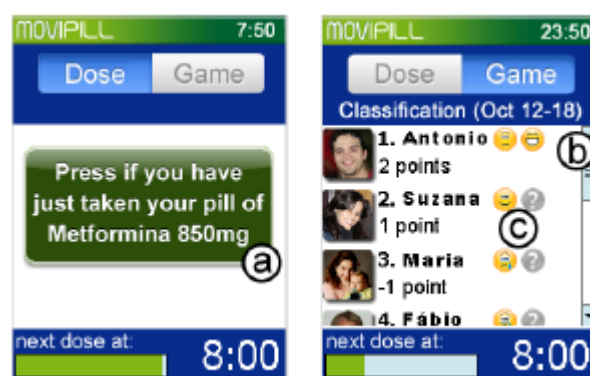


Figura 6. Imagem ilustrativa do *MoviPill* (de Oliveira et al., 2010)

O segundo caso é o *Waterbot* (Arroyo et al., 2005). O trabalho (observável na Figura 7) visa a exploração de interfaces do utilizador, interfaces de persuasão e técnicas de *feedback* referentes ao lavatório. Um dos aspetos relevantes foi terem referido técnicas de mudança comportamentais, onde para além do reforço positivo, reforço negativo e validação, há a retirar para o trabalho o automatismo, adição de valor e avisos em tempo real. Foram usadas como técnicas o reforço negativo, *feedback* e monitorização, bem como uma tentativa de aumento de consciência. O resultado final revelou que o *Waterbot* foi um sucesso, pois os utilizadores tiveram uma maior tendência em economizar recursos relativos ao lavatório. Além deste aspeto, os utilizadores apresentaram interesse na interface mesmo quando expostos a tempo prolongado, mostrando que é possível criar

mecanismos de persuasão baseados em contexto e num ambiente reduzido, com resultados positivos.



Figura 7. Imagem ilustrativa dos mecanismos do Waterbot (Visível mecanismo em que iluminam água de vermelho caso não esteja a ser usada corretamente). (Arroyo et al., 2005)

O terceiro trabalho é o *ViTo* (Nawyn, Intille, & Larson, 2006). É uma interface (visível na Figura 8) que promove discretamente a redução das horas a ver televisão enquanto encoraja um aumento na frequência e quantidade de atividades não sedentárias. A dificuldade dos autores passou por combater as atividades sedentárias, uma vez que essas atividades têm propriedades viciantes, como relaxamento e passividade (embora a longo prazo provoque uma sensação negativa no utilizador), enquanto pretendiam incentivar exercício físico que envolve compromisso e nem sempre os ganhos são visíveis a curto prazo. Outro desafio passou por não forçar o utilizador a escolher um lado, ou dizer o quão negativo é ver televisão, mas sim incutir esse pensamento (e que esse pensamento não seja só por justificação). A ideia passou por cortar a relação com a televisão e de seguida cortar os “custos” do exercício e torná-lo um reforço positivo. A solução para os desafios passou por um aparelho que fornece uma série de estratégias de mudança de atividades e comportamentos, com o objetivo de ajudar o utilizador a tomar decisões com um outro grau de informação. Este explorava *feedback* em tempo real e interação em formato jogo (observável na Figura 8), para promover a atividade física a um custo/impacto reduzido. Este também dava recompensas (reforço positivo) por atividades físicas bem como um aspeto de diversão. A quantidade e a qualidade das técnicas usadas para alterar o comportamento nesta aplicação, tornam a mesma um alvo de referência para posteriores aplicações.



Figura 8. Exemplo da aplicação ViTo. Lado Esquerdo - Menu Inicial; Lado Direito - as opções que o utilizador possui. (Nawyn et al., 2006)

Outros trabalhos foram estudados, onde embora o foco de estudo fosse a qualidade da interface, possuíam aspetos a ser retirados para o *PIQL*.

O *StoryVisit* (Raffle et al., 2011) permitia crianças e adultos a longa distância experimentarem uma sensação de junção através da leitura de livros de histórias de crianças à distância. O aspeto da inclusão social e de fazer atividades em conjunto são técnicas que podem ser utilizadas no *PIQL* (como referido na secção Técnicas de Alteração de Comportamentos, pelo técnica de Validação Social).

Em suma, com estas três secções, possuímos as bases para inserir no sistema *PIQL* um modelo de comportamento que explique ao máximo o comportamento dos utilizadores, técnicas que possam levar a mudança de comportamentos desviantes por parte do utilizador (e manter os seus comportamentos positivos), bem como a importância da ligação entre as técnicas e o sistema a ser utilizado. Estes aspetos serão futuramente abordados.

Capítulo 3

PersonAAL

O *PersonAAL*¹³ é o projeto base utilizado para a criação do sistema persuasivo *PIQL*. Este tem o objetivo de prolongar o tempo que as pessoas idosas podem viver no seu ambiente doméstico, aumentando a autonomia e a assistência em atividades que realizam diariamente. A plataforma *PersonAAL* é um *software* de suporte que permite aplicações *web*, focadas em idosos, serem aperfeiçoadas com mecanismos de adaptação e persuasão. O projeto *PIQL* será incluído na plataforma *PersonAAL*, consequentemente é necessário compreender a arquitetura do *PersonAAL*. Esta está apresentada na Figura 9, sendo composta por várias componentes, cada uma com funções específicas que asseguram o seu funcionamento correto.

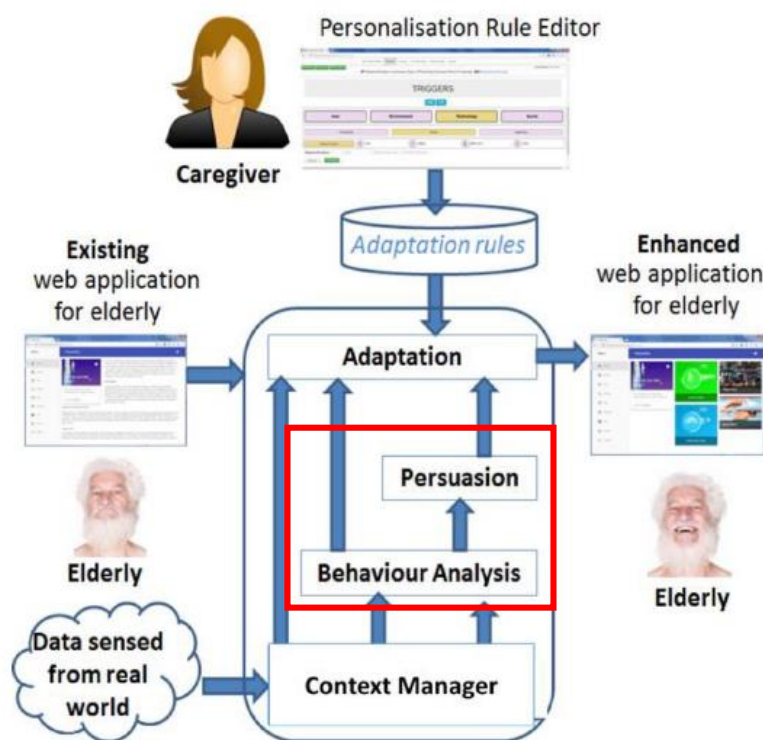


Figura 9. Plataforma *PersonAAL*. Vermelho – Localização do Sistema *PIQL*, que funcionará como Módulo de Análise de Comportamento e de Persuasão

¹³ Link onde se pode obter mais informações sobre o projeto *PersonAAL*, <http://www.personaal-project.eu/>

- O Módulo de Contexto (na Figura 9 descrito como *Context Manager*) recebe informação contextual de aplicações e dispositivos (referido na Figura 9 como *Data sensed from real world*) associados a cada utilizador (estas podem ser individuais ou embebidas no browser *web*), sendo estes denominados responsáveis de contexto. A informação de contexto é detetada pelos sensores dos responsáveis de contexto (por exemplo o responsável de contexto pela monitorização de atividades físicas depende da informação dada pelo acelerómetro).
- O Módulo de Análise Comportamental (referido na figura como *Behaviour Analysis*) está encarregado de monitorizar as atividades do indivíduo. O módulo possui *inputs* que originam a comparação entre o comportamento atual do utilizador e o comportamento que era esperado. Algumas das componentes do comportamento atual são enviadas para o módulo através do *Context Manager*. Posteriormente o Módulo de Análise Comportamental comunica com o Módulo de Adaptação, dado que o *output* da análise deve ser adaptado antes de ser entregue ao idoso. O módulo comunica com o Módulo de Persuasão, caso considere ser necessário o envio de uma mensagem persuasiva. Caso não haja necessidade da criação dessa mensagem, comunica diretamente com o Módulo de Adaptação.
- No Módulo de Persuasão (descrito na figura como *Persuasion*) ocorre a identificação de situações em que há necessidade de criar mensagens persuasivas a serem enviadas ao utilizador (quando o seu comportamento desvia-se do esperado ou reforçá-lo quando este é positivo). Esta necessidade é comunicada pelo Módulo de Análise Comportamental. A mensagem tem o intuito de mudar o comportamento negativo do utilizador ou tentar manter o seu comportamento positivo. Antes do envio da mensagem persuasiva, este módulo comunica com o Módulo de Adaptação.
- O Módulo de Adaptação (na figura descrito como *Adaptation* e de seu nome *Adaptation Engine*) fornece o suporte para adaptar as aplicações ao contexto de uso do utilizador. Este módulo permite receber regras de adaptação (*Adaptation Rules* na figura), dadas por uma ferramenta da responsabilidade dos cuidadores de saúde (na imagem referenciados como *Caregiver*), de forma a auxiliar a comunicação entre as aplicações e o utilizador. Consoante a informação do meio

onde o utilizador está inserido (fornecida pelo Módulo de Contexto) e através das regras de adaptação criadas, uma mensagem persuasiva (comunicada pelo Módulo de Persuasão) é enviada de forma personalizada ao utilizador. (Exemplificando: para a toma de medicamentos, é emitido um aviso sonoro. Se for detetada uma insuficiência auditiva no utilizador pode ser criada uma regra de adaptação onde o aviso começa a ser dado de uma forma diferente, neste caso, através de vibração de telemóvel ou aumento do nível sonoro do aviso). As regras de adaptação têm especificamente um *estímulo* e uma ação a realizar. As mudanças associadas às ações das regras despoletadas são transmitidas à aplicação em posse do cliente.

O projeto *PIQL*, como enunciado na legenda da Figura 9, será integrado no Módulo de Análise de Comportamento e no Módulo de Persuasão. O *Context Manager* fornecerá dados, aos módulos do *PIQL*, relacionados com informações e dados atuais do utilizador, já o Módulo de Adaptação estará encarregue de receber a mensagem persuasiva que será enviado pelo módulo de persuasão do *PIQL*.

Capítulo 4

Desenho

Neste capítulo é abordada a arquitetura do sistema *PIQL*. Serão discutidas as decisões tomadas desde o conceito inicial até ao seu desenho final. Também serão identificados os vários componentes associados ao sistema *PIQL* bem como descrições e aspetos peculiares dos mesmos.

4.1 Conceptualização

O sistema *PIQL*, como referido na secção Introdução, visa a exploração e desenho de interfaces persuasivas, com o intuito de incentivar os utilizadores a adotarem estilos de vida saudáveis, quer do ponto de vista físico como do ponto de vista cognitivo. O sistema será inserido no contexto do *PersonAAL*, incorporado no Módulo de Análise de Comportamento e no Módulo de Persuasão onde o objetivo será a criação de uma mensagem persuasiva com a intenção de modificar comportamentos considerados negativos e reforçar os comportamentos positivos do utilizador.

Atendendo ao conceito do sistema, o passo inicial será a construção de um módulo que permita a recolha de dados do utilizador. Posteriormente através de um mecanismo ou algoritmo, os dados são processados de modo a obtermos padrões de comportamento do utilizador.

Obtidos os padrões do utilizador, o passo seguinte é a criação de um módulo que obtenha a informação relacionada com o comportamento atual do utilizador. Esta informação será comparada com a informação recolhida anteriormente relativa aos padrões, tornando possível a deteção de desvio destes últimos. O desvio pode ter um aspeto positivo, em que o comportamento do utilizador atualmente é melhor comparativamente com o esperado, um aspeto negativo caso o comportamento atual seja pior que o comportamento recolhido dos padrões, ou pode não haver uma deteção de desvio de comportamentos.

Um terceiro módulo deve ser responsável por obter o tipo de desvio apresentado no módulo anterior. O módulo, através de um modelo de comportamento, irá identificar fatores que expliquem o comportamento que o utilizador exhibe. Caso haja um desvio de

comportamento, o módulo indica quais os fatores que registaram mudanças nos seus valores, indiciando os possíveis fatores que levaram à alteração de comportamento por parte do utilizador.

Registados os fatores que podem vir a explicar a mudança de comportamento, um quarto módulo indica quais as técnicas persuasivas que melhor abordam esses fatores e consequentemente, vir a alterar ou reforçar o comportamento do utilizador.

Identificadas as técnicas persuasivas que possivelmente permitem ao utilizador melhorar o seu comportamento ou sustenta-lo, estas técnicas são transformadas em mensagens persuasivas e posteriormente são aglomeradas numa única mensagem persuasiva a ser enviada para o utilizador. Esta mensagem será posteriormente enviada para o Módulo de Adaptação do *PersonAAL*

A Figura 10 evidência o esquema dos módulos a serem criados pelo *PIQL*, os quais foram explicados anteriormente, onde os três primeiros estão inseridos no Módulo de Análise de Comportamento do *PersonAAL* e os dois últimos incorporados no Módulo de Persuasão.

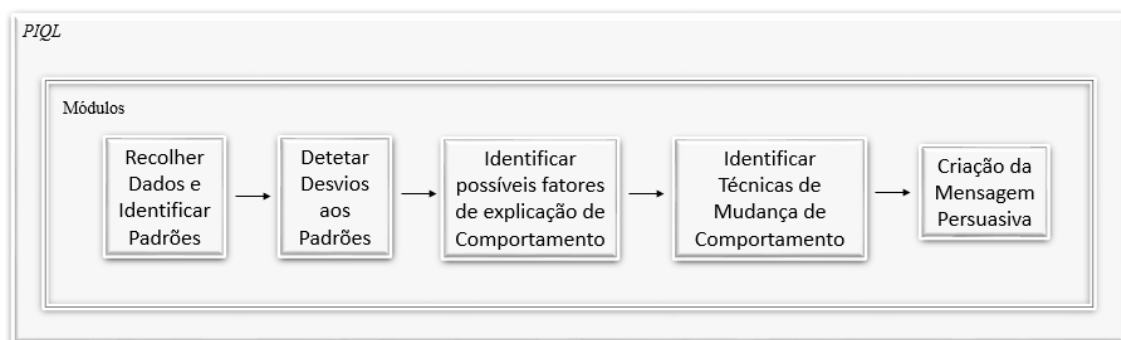


Figura 10. Esquema representativo do conceito inicial do sistema *PIQL*.

4.2 Módulos do *PIQL*

A Figura 11 representa o desenho atual do sistema *PIQL*. Na figura é possível observar a correlação entre os módulos iniciais e atuais (por exemplo o Módulo de recolha de dados e identificação de padrões da Figura 10 corresponde ao Módulo *IdentifyPatternsFinal* da Figura 11) bem como as ligações existentes entre cada módulo e as componentes que o auxiliam.

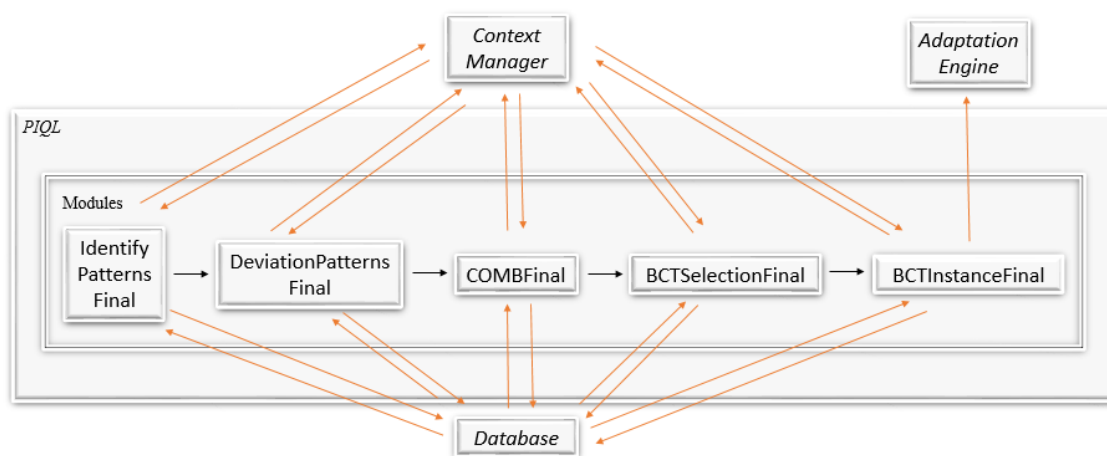


Figura 11. Esquema representativo da conversão do conceito inicial (Figura 10) para os módulos atuais do PIQL.

4.2.1 Identificação de Padrões

O módulo inicial do *PIQL* denomina-se *IdentifyPatternsFinal* e corresponde ao módulo de recolha de dados e identificação de padrões. O tipo de dados que serão recolhidos e posteriormente analisados para a criação de padrões provêm da atividade física do utilizador. Esta decisão foi tomada baseada no parecer dos parceiros europeus. A atividade física provém do *Context Manager*, o qual tem guardado os dados sobre todos os utilizadores. Dado que a recolha de dados não é da responsabilidade do *PIQL*, derivado da falta de dados, usou-se para o desenvolvimento do módulo um *dataset* disponível com atividade física. No *dataset* de Meyer (Meyer, Tran, & Greenwood, 2016) está registado o número de passos dados por um utilizador em intervalos de 15 minutos, durante dois meses (possível observar o excerto no Apêndice A – Ficheiro Excel de Atividade Física). A atividade física registada por parte do *dataset* foi inserida na base de dados do projeto *PIQL* para efeitos de teste. O módulo encarrega-se de retirar os dados do *Context Manager* (embora como indicado anteriormente, o módulo retira a informação da base de dados para casos de teste) relativos à informação registada do utilizador, e através do algoritmo de Meyer, cria-se padrões de atividade.

O algoritmo de *Meyer* (Meyer et al., 2016) usa a distância entre dois dias calculada pela fórmula:

$$D_{ii'} = \min_{j \in [-8, \dots, 8]} \left\{ \frac{\sum_{t=1}^{96} |X_{i[t+j]} - X_{i't}|}{96-|j|} \right\}$$

O algoritmo de *Meyer* (Meyer et al., 2016) compara dias de atividade física do utilizador, sendo os dias posteriormente adicionados em grupos (visível na Figura 12). Observando a fórmula do algoritmo de *Meyer*, este calcula a distância entre dois dias (representado por $D_{ii'}$). Dias que possuam distâncias semelhantes serão agrupados no mesmo grupo, criando assim padrões de atividade. O cálculo da distância entre dias (sendo os dias i e i') é feito através da diferença de passos (passos estes representados por X) entre intervalos correspondentes dos dois dias (sendo cada intervalo representado por t).

$$|X_{i[t+j]} - X_{i't}|$$

Dado que cada dia possui 96 intervalos de 15 minutos, será necessário realizar a diferença de todos os 96 intervalos e somar a diferença de passos em cada intervalo.

$$\sum_{t=1}^{96} |X_{i[t+j]} - X_{i't}|$$

De forma a obtermos um valor médio da distância de passos, divide-se pelo número de intervalos comparados.

$$\frac{\sum_{t=1}^{96} |X_{i[t+j]} - X_{i't}|}{96-|j|}$$

O algoritmo possui uma característica que permite adiantar ou atrasar os intervalos de passos dos dias a serem comparados, de forma a haver uma comparação mais exata entre os registos de atividade dos dois dias (exemplificado na Figura 13 e representado no algoritmo pela variável j). O limite para adiantar ou atrasar intervalos dos dias vai até 2 horas antes e depois de intervalos correspondentes. Com o valor de j a 0, é calculado intervalos correspondentes, por exemplo passos do 1º intervalo do 1º dia com os passos do 1º intervalo do 2º dia). Se for necessário adiantar a atividade do utilizador para 2 horas, o valor de j será 8 (cada valor numérico de j corresponde a 15 minutos de atividade, onde 8 corresponde a 120 minutos, 2 horas). Nesse caso será comparado o 9º intervalo do 1º dia com o 1º intervalo do 2º dia, 10º com o 2º, até chegar ao 96º intervalo.

$$j \in [-8, \dots, 8] \left\{ \frac{\sum_{t=1}^{96} |X_{i/t+j} - X_{i't}|}{96-|j|} \right\}$$

O valor da distância dos dois dias será o valor mínimo registado pelo algoritmo, comparando a atividade dos dois dias com a janela de 2 horas antes até 2 horas depois do momento que devia ser comparado.

$$D_{ii'} = \min_{j \in [-8, \dots, 8]} \left\{ \frac{\sum_{t=1}^{96} |X_{i[t+j]} - X_{i't}|}{96-|j|} \right\}$$

Com esta informação é possível criar grupos de dias, visível na Figura 12. Através dos grupos de dias que foram criados é possível obter informação dos dois grupos mais distintos (por exemplo número médio de passos dados em cada intervalo de 15 minutos ou número de dias que cada grupo possui). Este tipo de informação, detalhado futuramente na secção Implementação, corresponde aos padrões de atividade do utilizador e são guardados na base de dados do *PIQL*.

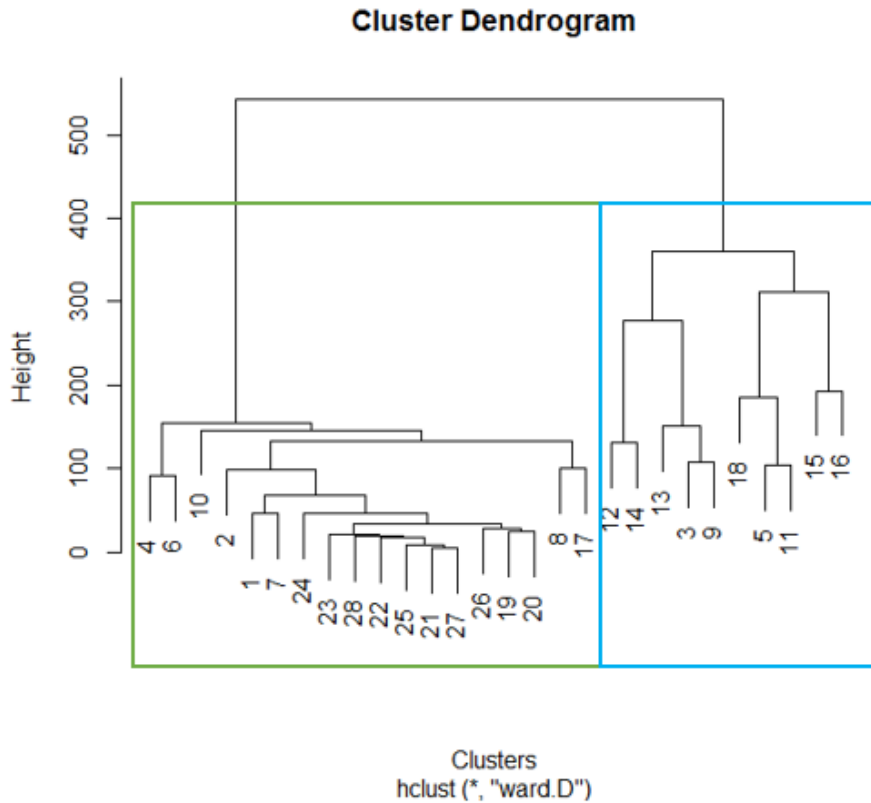


Figura 12. Dendrograma que mostra os dois distintos grupos de atividade baseados nos últimos 28 dias de atividade do utilizador.

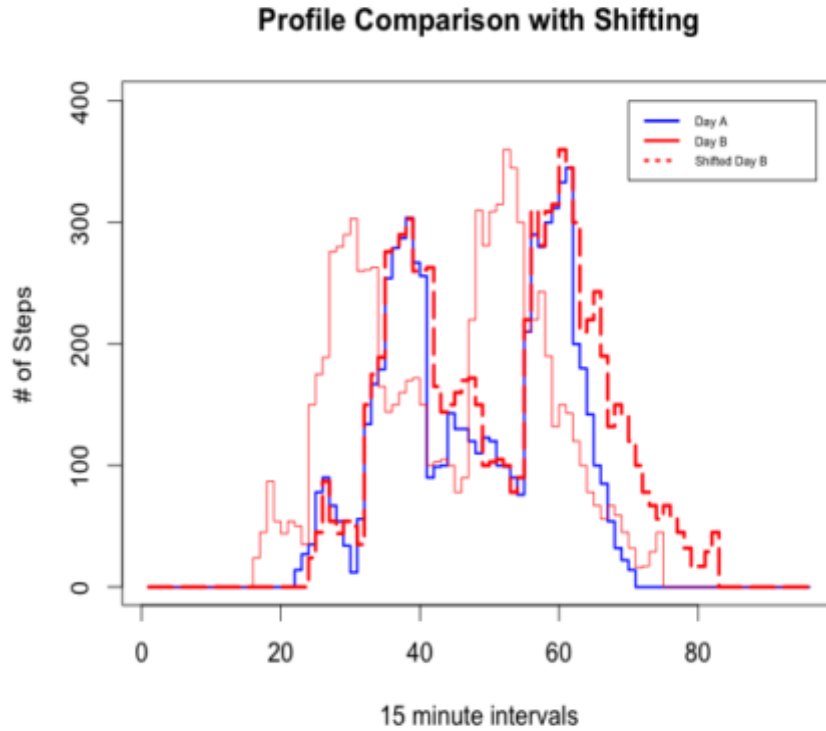


Figura 13. Comparação entre dois dias (Dia A a azul e B a vermelho). O Dia B (a vermelho) foi adiantado 2 horas (representado a vermelho tracejado) para realizar uma comparação mais semelhante entre os dois dias.

4.2.2 Detecção de desvios

O módulo que deteta o desvio dos padrões é denominado *DeviationPatternsFinal*. Este módulo recebe do *Context Manager* (para efeitos de desenvolvimento e teste recebe da base de dados) as informações relacionadas com os padrões de atividade do utilizador. Recebe também o registo da atividade diária realizada pelo utilizador, a qual é processada posteriormente, de forma a possibilitar a comparação com a informação dos padrões. O primeiro passo de comparação é compreender em qual dos grupos de padrão o novo dia se enquadra. Após ter-se conhecimento do grupo a que este pertence, compara-se a informação entre ambos. Para tal foram criadas cinco aproximações (visíveis na Tabela 7 e explicadas na secção 4.2.2.1) em que cada uma usa informações distintas que permitem analisar se a atividade física do novo dia se desvia do seu grupo de padrão, designado *cluster*, e qual o tipo de desvio (positivo ou negativo). Das cinco aproximações, a quinta é a que possui melhores resultados, onde prevê com sucesso 91% dos casos que possuem desvio.

Aproximações existentes no Módulo DeviationPatternsFinal	
Tipo de Aproximação	Explicação
Aproximação 1	<p>Através dos dados do padrão, onde o novo dia está inserido, é criada a média de passos dada por intervalo de tempo (15 minutos). Calcula-se de seguida um limite superior e inferior para essa média. Calcula-se para o novo dia a média de passos dados. Caso esse valor ultrapasse um dos limites, considera-se que há desvio.</p>
Aproximação 2	<p>Através dos dados do padrão, onde o novo dia está inserido, é calculada a energia diária despendida pelo utilizador (originada pelo número de passos que este realiza). A partir deste valor são criados limites inferiores e superiores de energia. Caso o valor de energia do novo dia não esteja situado entre o intervalo dos limites, considera-se que há desvio.</p>
Aproximação 3	<p>Através dos dados do padrão, onde o novo dia está inserido, é criado para cada intervalo (15 minutos) um limite superior e inferior de passos que podem ser dados. Para cada intervalo do novo dia é verificado se o número de passos deste está dentro do intervalo. Caso o número de intervalos onde os limites foram ultrapassados seja superior a um dado número, considera-se que há desvio.</p>
Aproximação 4	<p>Através dos dados do padrão, onde o novo dia está inserido, é calculada a energia diária despendida pelo utilizador a cada intervalo de tempo. A partir deste valor são criados limites inferiores e superiores de energia. É realizado o</p>

	mesmo cálculo de energia para os intervalos do novo dia. Caso o valor total de energia dos intervalos onde os limites foram ultrapassados seja superior a uma constante, considera-se que houve desvio.
Aproximação 5	Esta aproximação recebe todos os dados relativos às aproximações anteriores, bem como outras informações relativas ao novo dia. Com estas informações e com classificações de casos de testes realizados, é criada uma árvore de decisão que determina o tipo de desvio que ocorreu.

Tabela 7 – Resumo das aproximações usadas para detetar o desvio no Módulo *DeviationPatternsFinal*

4.2.2.1 Pseudo-código e fórmulas das aproximações

Antes de serem processadas as aproximações, é necessário sabermos a que grupo de atividade, também designado *cluster*, o novo dia pertence. Como tal os passos são:

Obter o valor dos passos dados em cada intervalo de 15 minutos do novo dia

Obter o valor dos passos dados em cada intervalo de 15 minutos dos diferentes dias presentes nos diferentes *clusters*

Calcular para cada *cluster* a média de passos dados em cada intervalo de 15 minutos (será designado dia representativo do *cluster*).

Usar o algoritmo de *Meyer* para calcular a distância entre o dia representativo do 1º *cluster* e o novo dia, bem como do dia representativo do 2º *cluster* com esse mesmo dia. O dia representativo é a média dos dias que se situam no *cluster*.

Obtidos esses resultados, são comparadas as duas distâncias. A que for menor representa o *cluster* a que o novo dia pertence. Registada esta informação, podem-se executar as aproximações que vão detetar se existe um desvio.

Aproximação 1

É necessário para esta aproximação obter:

- Média da distância entre cada dia do *cluster*, onde o novo dia está inserido, e o dia representativo do *cluster*, com o auxílio do algoritmo de *Meyer* (o valor será designado D (DiaRepresentativo)).
- Desvio padrão das distâncias (designado DesvioPadrão).
- Distância do dia representativo do *cluster* ao dia mais afastado do centro do *cluster* (designado D (DiaAfastado)).

Obtidos estes dados é usada a fórmula de *Meyer* para obter a distância entre o novo dia e o dia representativo, que será representada como D (NovoDia). De seguida é calculado se existe um desvio grave, simples ou nulo.

Existe desvio grave caso:

$$D(\text{NovoDia}) > D(\text{Dia Afastado}) * n, \text{ sendo } n \text{ uma constante de refinação.}$$

Existe desvio pequeno caso:

$$D(\text{NovoDia}) > D(\text{DiaRepresentativo}) + (\text{DesvioPadrão} * y), \text{ sendo } y \text{ uma constante de refinação}$$

Caso contrário não existe desvio.

Através destas fórmulas é possível calcular a gravidade do desvio ou a ausência do mesmo. Caso haja desvio averigua-se o seu tipo (se positivo ou negativo):

Se existir desvio (alguma das fórmulas acima referidas se verificar)

Caso Passos do novo dia > Passos do dia representativo

Existe desvio positivo

Caso Passos do novo dia < Passos do dia representativo

Existe desvio negativo

Não existe desvio

Desvio nulo

Aproximação 2

É necessário para esta aproximação obter:

- A energia usada em média por dia em cada *cluster* (representado por E (DiaCluster)). A energia de um dia é representada pela fórmula:

$$E = \sum_{i=1}^{96} S_i^2$$

S_i é o número de passos dados no intervalo i . Energia é o somatório do quadrado de cada intervalo de 15 minutos. Calcula-se a energia do dia representativo dos *clusters* através da fórmula de energia, obtendo o E (DiaCluster)

- Desvio padrão das Energias (representado por DesvioPadrão).

Obtidos estes dados, calcula-se a energia do novo dia através da fórmula da energia, o qual será representado por E (NovoDia). De seguida é calculado se existe desvio grave, pequeno ou nulo.

Existe desvio grave caso:

E (NovoDia) $>$ E (DiaCluster) + DesvioPadrão * m , sendo m uma constante de refinação.

Ou

E (NovoDia) $<$ E (DiaCluster) - DesvioPadrão * m , sendo m uma constante de refinação.

Existe desvio pequeno caso:

E (NovoDia) $>$ E (DiaCluster) + DesvioPadrão * p , sendo p uma constante de refinação.

Ou

E (NovoDia) $<$ E (DiaCluster) - DesvioPadrão * p , sendo p uma constante de refinação.

Onde $m > p$. Através destas fórmulas é possível calcular a gravidade do desvio ou a ausência do mesmo. Caso haja desvio averigua-se o seu tipo (se positivo ou negativo):

Se existir desvio (se verificar algumas das fórmulas acima referenciadas)

Caso $E(\text{NovoDia}) > E(\text{DiaRepresentativo})$

Existe desvio positivo

Caso $E(\text{NovoDia}) < E(\text{DiaRepresentativo})$

Existe desvio negativo

Não existe desvio

Desvio nulo

Aproximação 3

É necessário para esta aproximação obter:

- Valor médio de passos de cada intervalo de 15 minutos do *cluster* em que se situa o novo dia.
- Desvio padrão de cada intervalo de 15 minutos do dia representativo do *cluster* em que se situa o novo dia.

Obtendo estes valores, para cada intervalo de 15 minutos, é criado um limite superior e um limite inferior.

Limite superior para cada intervalo de 15 minutos:

Valor médio de passos do cluster nesse intervalo + desvio padrão nesse intervalo * u, sendo u uma constante de refinação.

Limite inferior para cada intervalo de 15 minutos:

Valor médio de passos do cluster nesse intervalo - desvio padrão nesse intervalo * d, sendo d uma constante de refinação.

De seguida é usado o seguinte algoritmo:

Criar 2 variáveis de contagem, v1 e v2

Para cada intervalo de 15 minutos

Comparar o valor de passos do intervalo do novo dia com o valor de passos dos limites

Caso o valor de passos do intervalo ultrapasse o limite superior

Incrementar $v1$

Caso o valor de passos do intervalo esteja abaixo do limite inferior

Incrementar $v2$

Soma-se $v1$ com $v2$, criando lim , que é o número de vezes total que os limites foram ultrapassados.

Caso $lim \geq m$, sendo m um número máximo de vezes que os limites podem ser ultrapassados para ser considerado desvio grave

Existe desvio grave

Caso $lim \geq n$, sendo n um número máximo de vezes que os limites podem ser ultrapassados para ser considerado desvio pequeno bem como $n < m$

Existe desvio pequeno.

Caso contrário não há desvio

Em termos de tipo de desvio:

Caso $lim < n$

Não há desvio

Caso contrário

Se $v1 > v2$

Há desvio positivo

Se $v2 > v1$

Há desvio negativo.

Aproximação 4

É necessário para esta aproximação obter:

- Valor de passos de cada intervalo de 15 minutos do dia representativo do *cluster* em que se situa o novo dia.
- Desvio padrão de cada intervalo de 15 minutos do dia representativo do *cluster* em que se situa o novo dia.

Obtendo estes valores, para cada intervalo de 15 minutos, é criado um limite superior e um limite inferior.

Limite superior para cada intervalo de 15 minutos:

Valor de passos do dia representativo nesse intervalo + desvio padrão nesse intervalo * u, sendo u uma constante de refinação.

Limite inferior para cada intervalo de 15 minutos:

Valor de passos do dia representativo nesse intervalo - desvio padrão nesse intervalo * d, sendo d uma constante de refinação.

De seguida é usado o seguinte algoritmo:

Criar 2 variáveis de contagem, c1 e c2

Para cada intervalo de 15 minutos

Comparar o valor de passos do intervalo do novo dia com o valor de passos dos limites

Caso o valor de passos do intervalo ultrapasse o limite superior

$c1 = c1 + (\text{valor de passos do novo dia} - \text{valor de passos do limite superior})$

Caso o valor de passos do intervalo esteja abaixo do limite inferior

$c2 = c2 + (\text{valor de passos do novo dia} - \text{valor de passos do limite superior})$

Soma-se v1 com v2

Calcula-se o somatório de todos os intervalos de 15 minutos do novo dia, registado como T

Calcula-se $((c1+c2) * 100) / T$, obtendo uma percentagem denominada per .

Obtendo o per, pode-se calcular o desvio.

Se $per > m$, sendo m uma percentagem relativa à percentagem de desvio grave

Existe desvio grave

Se $per > p$, sendo p uma percentagem relativa à percentagem de desvio pequeno e a qual é menor que m

Existe desvio pequeno

Caso contrário, não há desvio

Em termos de desvio

Se $c1 > c2$

Existe desvio positivo

Se $c2 > c1$

Existe desvio negativo

Aproximação 5

A Aproximação recebe dados do módulo anterior, cria dados através dos algoritmos das outras aproximações e cria novas variáveis na aproximação 5 de forma a possuir dados para criar uma árvore de decisão.

Do módulo anterior:

- Distância entre dias representativos dos *clusters*.

Dados criados das outras aproximações:

Aproximação 1

- Distância entre o novo dia e o dia representativo do seu *cluster*, D (NovoDia).
- Distância entre dia representativo do *cluster* do novo dia e o dia mais afastado, D (DiaAfastado).

Aproximação 2

- Energia do novo dia, através da fórmula de energia E (Dia).
- Energia média gasta por dia no *cluster*, E (DiaRepresentativo)

Aproximação 3

- Número de vezes que o limite superior foi ultrapassado, v1.
- Número de vezes que o limite inferior foi ultrapassado, v2.

Aproximação 4

- Número de passos acima do limite superior da aproximação 4, c1
- Número de passos abaixo do limite inferior da aproximação 4, c2

Novas variáveis criadas na própria aproximação:

- Energia do dia mais afastado pertencente ao *cluster* do novo dia.
- Passos dados entre as 0 e as 6 horas do novo dia.
- Passos dados entre as 6 e as 12 horas do novo dia.
- Passos dados entre as 12 e as 18 horas do novo dia.
- Passos dados entre as 18 e 24 horas do novo dia.

Juntamente a estes dados, foi introduzida a classificação que cada dia aparentava ter. Os dias foram dados através de casos de testes realizados a partir do *dataset* de Meyer (exemplificado na Figura 14). A classificação consistiu na avaliação dos orientadores e mentorando, em que cada indivíduo analisava individualmente a atividade física de cada dia comparativamente com a média do seu *cluster* (semelhante à Figura 14). Para cada dia era anotada as classificações dadas pelos avaliadores. A classificação do dia com mais votos era a introduzida. Em caso de empate (os três avaliadores possuírem pareceres distintos) era debatido em conjunto a avaliação do dia até chegar a um consenso.

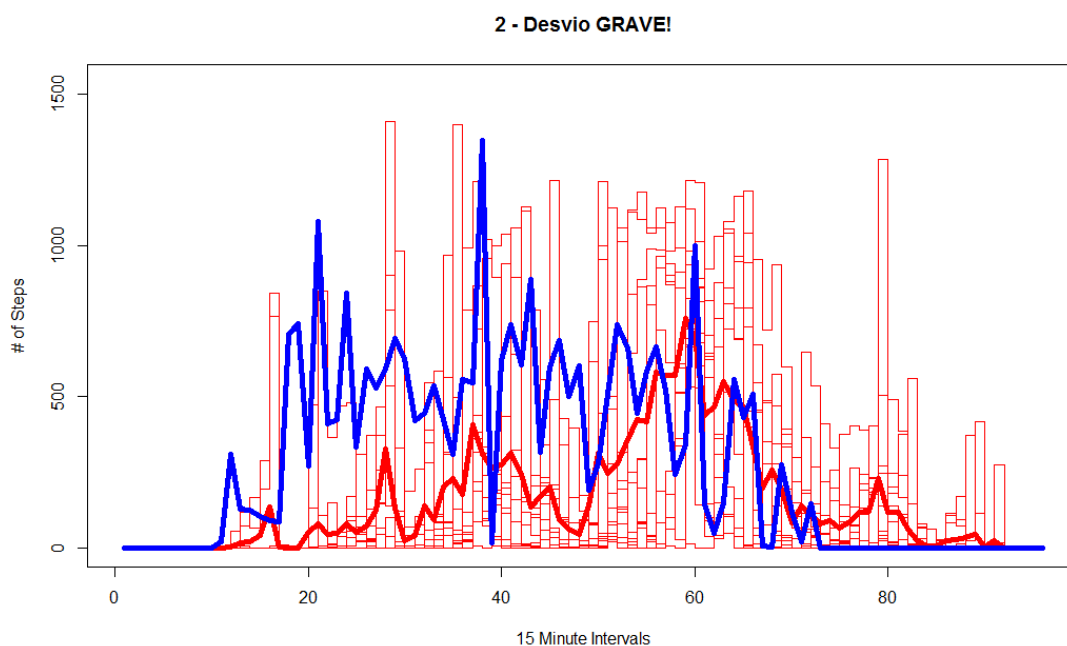


Figura 14. Um dos casos de teste avaliado. A vermelho – Dia representativo do cluster. A azul – Novo dia. Pela inconsistência da atividade e desvio comparativo ao dia representativo, este dia é classificado de desvio grave.

Obtidos os dados, construiu-se uma árvore de decisão que teve como objetivo prever qual o desvio feito nesses dias. Dos 33 casos de teste, 30 obtiveram resultados coincidentes com a previsão, sendo que 3 registaram uma divergência de resultados no desvio, registrando a classificação mais elevada das 5 aproximações, com uma taxa de sucesso de 91%.

A percentagem de sucesso para cada uma das aproximações está evidenciada na Tabela 8.

Tipo de Aproximação	Taxa de Sucesso
Aproximação 1	55%
Aproximação 2	76%
Aproximação 3	85%
Aproximação 4	73%
Aproximação 5	91%

Tabela 8 – Taxa de Sucesso em cada aproximação

4.2.3 Modelação de comportamento

O módulo encarregue de identificar possíveis fatores que expliquem o comportamento do utilizador é denominado de *COMBFinal*. Estes fatores provêm do modelo de comportamento escolhido, sendo o COM-B selecionado. A decisão começou a ser tomada no momento em que foram analisadas as vantagens do COM-B relativamente aos restantes modelos (assinalados no Capítulo “Trabalho Relacionado”), tais como o reconhecimento do comportamento como um sistema interativo, o delinear simples e fácil das etapas e a relação entre elas, o permitir identificar objetivos (dando espaço para personalizar a aplicação) e a facilidade em criar formas de corrigir o problema. Embora a sugestão de modelo de comportamento fosse esta, a validação para que este fosse usado veio de Jan Nordvik, psicólogo envolvido no projeto *PersonAAL*, onde de todos os modelos estudados, aconselhou fortemente o uso do COM-B. Além dessa contribuição, através de reuniões e discussão, foi possível chegar a uma lista de fatores que podiam explicar desvios de comportamento (referidos no Apêndice B – Lista de Fatores que afetam comportamento). Os fatores são ao nível da capacidade, motivação e oportunidade, tendo sido estes registados na base de dados. O módulo recebe da base

de dados o tipo de desvio calculado no módulo anterior, e consoante os fatores registados na base de dados, verifica quais deles possuem dados e que possam explicar o tipo de desvio. Por exemplo, o facto de o tempo ter estado bom, criou a oportunidade para o utilizador caminhar e criar o desvio positivo que foi gerado. Contrariamente se o desvio foi considerado negativo, o facto de o tempo local ser péssimo pode explicar a diminuição dos níveis de atividade física. Os possíveis fatores que possam explicar o desvio calculado são utilizados pelo módulo seguinte.

4.2.4 Seleção de técnicas de alteração de comportamentos

O quarto módulo, indicado na Figura 10 como o módulo que identifica técnicas de mudança de comportamentos, é o correspondente ao *BCTSelectionFinal*. As técnicas de mudança de comportamento estudadas na secção “Técnicas de Alteração de Comportamentos”) foram debatidas com o psicólogo Jan Nordvik, o qual sugeriu algumas mudanças, originando a lista de Técnicas de Alteração de Comportamentos (denominadas BCT’s ou *Behaviour Change Techniques* ou técnicas persuasivas, registadas no Apêndice C – Lista de Técnicas de Persuasão). O módulo recebe os fatores que podem explicar o desvio de comportamento, calculados no módulo anterior, e verifica os valores associados a esses fatores (através do auxílio do Módulo de Contexto do *PersonAAL*). Caso existam fatores sem valores atribuídos, estes são descartados. Dos restantes fatores, são analisadas regras que verificam realmente quais os fatores que explicam o desvio. Caso uma regra seja ativada, são invocadas as técnicas de mudança de comportamento relativas a essa regra, de forma a combaterem os desvios negativos ou a reforçar os desvios positivos registados. Um exemplo: é detetado um desvio negativo e sabe-se que o tempo local pode levar o utilizador a diminuir a sua atividade física devido a esse fator. A regra avalia se o tempo está mau, e caso esteja, a técnica de comportamento associada a esta regra é a “resolução de problemas”, que irá passar pelo envio de uma mensagem a dizer ao utilizador que pode fazer atividades dentro de casa. Para cada fator são avaliadas as regras que lhes estão associadas, as quais podem invocar uma ou mais técnicas persuasivas. As técnicas invocadas a partir das regras bem como os fatores que explicam o desvio do comportamento são utilizadas no último módulo.

4.2.5 Implementação de técnicas de alteração de comportamento

O quinto e último módulo, que corresponde à criação da mensagem persuasiva é denominado *BCTInstanceFinal*. O módulo recebe os fatores que explicam o desvio de comportamento bem como as técnicas de mudança de comportamento associadas a esses fatores. A cada fator e técnica está associada uma mensagem. O módulo retira da base de dados a mensagem persuasiva indicada para cada fator e técnica de persuasão, onde posteriormente aglomera-as, obtendo a mensagem de persuasão que deve ser enviada para o utilizador. Caso existam técnicas repetidas, são retiradas da base de dados as mensagens correspondentes, sendo sorteada uma delas para pertencer à mensagem persuasiva final. A mensagem é depois enviada para o Módulo de Adaptação, de forma a ser recebida pelo utilizador consoante o seu contexto (por exemplo se o utilizador for cego, a mensagem pode-lhe ser enviada a partir de um sinal sonoro) e esta é guardada no *Context Manager* (em termos de desenvolvimento e testes, foi guardada na base de dados), de forma a ter um registo das mensagens enviadas para cada utilizador.

Capítulo 5

Implementação

Neste capítulo são abordadas escolhas feitas em termos de implementação do sistema *PIQL*.

5.1 Ferramentas utilizadas

Esta secção analisa as ferramentas utilizadas, tais como o ambiente de desenvolvimento do projeto, linguagem do projeto e estruturas auxiliares.

Em termos de ambiente de desenvolvimento integrado, a escolha recaiu no *Eclipse*, embora em fases iniciais tenha sido utilizado *IntelliJ Community Edition*, sendo que o projeto consegue ser executado em ambos os ambientes. O *IntelliJ* possui vantagens evidentes comparativamente com o *Eclipse*, tais como a indexação, *debugging*, sistema de autocorreção e refactoração. Atendendo às vantagens enunciadas, o *IntelliJ* é a escolha certa, porém uma das suas desvantagens é que usa mais memória que o *Eclipse*, sendo este ponto fulcral para a escolha feita, pois a performance do computador onde o sistema foi desenvolvido impossibilita o *IntelliJ* correr ao nível das capacidades que este possui, chegando a ser bem mais lento em altura de desenvolvimento que o *Eclipse*. Atendendo a esta desvantagem, optou-se pelo ambiente de trabalho *Eclipse*. Este possui a vantagem de ter uma larga comunidade bem como uma maior variedade de documentação. A versão usada do *Eclipse* foi o *Eclipse Jee Neon*.

A linguagem utilizada foi Java, sendo o Java Development Kit (JDK) a versão 1.8. A medida foi tomada tendo em conta a vasta documentação disponível para esta linguagem, a capacidade de se adaptar a múltiplos sistemas operativos e permitir o uso de múltiplas *frameworks* e *plugins* associados.

O projeto é auxiliado por *gradle*. *Gradle* é um sistema de gestão de compilação que usa conceitos de outras ferramentas de compilação (*Apache Ant* e *Apache Maven*) e foi desenhado para compilar projetos que podem tornar-se enormes, ao reconhecer quais as componentes que devem ser compiladas e evitar recompilar componentes que não foram modificadas desde a última compilação. A decisão foi tomada baseada em vários factos. Um dos principais fatores foi a facilidade em gerir dependências. Com o *gradle* temos acesso ao repositório *Maven*, onde pode-se facilmente retirar as dependências necessárias para a execução do projeto. Esta facilidade prende-se também por este processo ser feito

através de um ficheiro (*build.gradle*), o qual é de fácil programação e bastante legível. *Gradle* possui também *plugins* os quais se revelam benéficos para este tipo de projeto. Permite a criação de cápsulas, as quais possuem todas as dependências e classes do projeto, possibilitando a execução do projeto noutros ambientes. Este tipo de solução permitiu passar o projeto, preso num ambiente de desenvolvimento integrado, para o servidor *web*, tudo através de um ficheiro *.jar*. De forma a integrar o projeto *gradle* no *Eclipse*, foi necessário o uso do *Eclipse Marketplace* e instalar o *plugin BuildShip*¹⁴, obtendo o aspeto visível na Figura 15.

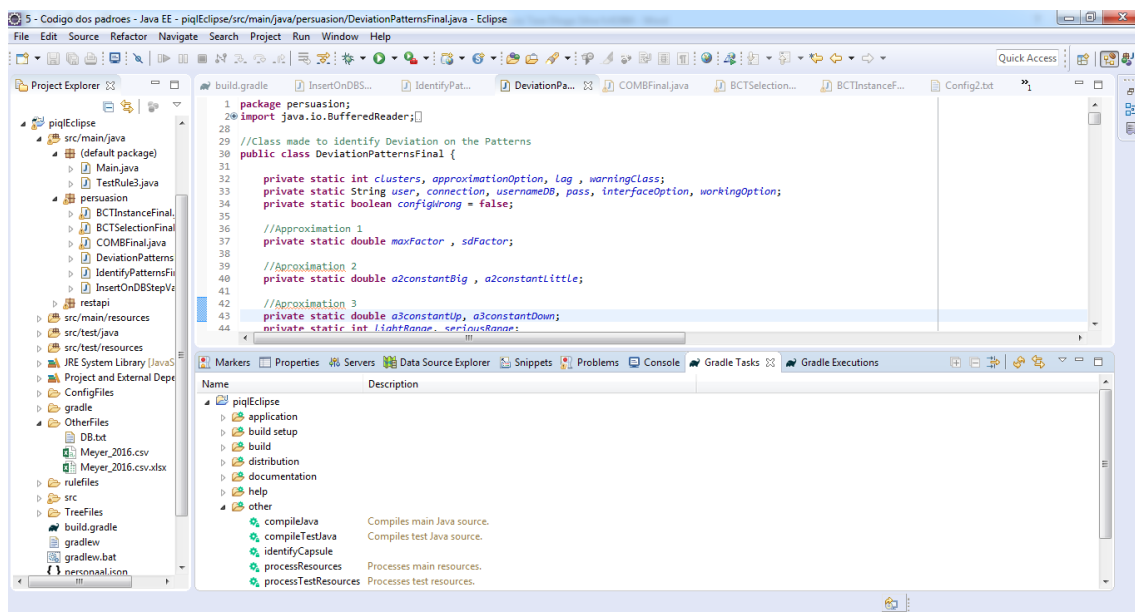


Figura 15. Aspeto do projeto *gradle* acerca do sistema *PIQL* no ambiente de desenvolvimento integrado *Eclipse*

O projeto possui também como estrutura auxiliar uma base de dados, que permite guardar estados e resultados que são originados em cada módulo existente no sistema *PIQL*.

Adicionalmente, o projeto requer uma plataforma *R*. Esta possibilita a criação de um servidor que permite receber pedidos feitos em *java* e processá-los em *R* conjuntamente com as suas bibliotecas, retornando o resultado na linguagem *java*. Esta necessidade provém da implementação realizada no 1º módulo, a qual será detalhada futuramente.

¹⁴ Passos de instalação em <http://www.vogella.com/tutorials/EclipseGradle/article.html>. Observado a 3 de Julho de 2017

O projeto possui um servidor onde ficará hospedado o projeto *PIQL* e será possível executar os diferentes módulos nas suas diferentes configurações. A ponte desde a criação das cápsulas até à sua execução no servidor é feita através de um repositório GIT, o qual permite sincronizar conteúdo entre o computador onde o sistema é desenvolvido e o local no servidor onde será possível executá-lo.

5.2 Módulos

Como explicado na secção Desenho, o sistema *PIQL* possui cinco módulos, cada um com características específicas e o mais independentes possível entre eles, para que caso seja necessária uma mudança num dos módulos, esta não afete todos os outros.

Embora não seja um dos cinco módulos oficiais, foi criado um módulo inicial extra, de seu nome *InsertOnDBStepValues*. Esta decisão foi tomada para caso seja necessário a inserção de valores de atividade física na base de dados para um dado utilizador, haver um módulo que crie facilmente esse aspeto. O módulo possui um ficheiro de configuração denominado *insertDBConfig.txt* que possui o utilizador, o nome do ficheiro Excel com os dados sobre a atividade física e a data de início da atividade física. É necessário ter um ficheiro Excel com esse nome (observável no Apêndice A – Ficheiro Excel de Atividade Física). O módulo retira os passos dados pelo utilizador a cada 15 minutos, e acendendo à base de dados, insere na tabela *StepCount*, os passos do utilizador referido no ficheiro de configuração, a cada 15 minutos, com a data de início mencionada no ficheiro.

IdentifyPatternsFinal é o módulo inicial do sistema *PIQL*, o qual está encarregue da identificação de padrões de atividade por parte do utilizador. Este módulo foi baseado no conceito de que entre todo o tipo de atividade que o utilizador vai realizar, a que mais nos interessa é a atividade física derivado do facto que esta foi a atividade escolhida pelos parceiros europeus. A atividade física no sistema *PIQL* é definida pelo número de passos dados pelo utilizador em intervalos de 15 minutos, como referido na secção Identificação de Padrões (possível observar um excerto através do Apêndice A – Ficheiro Excel de Atividade Física), estando estes guardados na base de dados. O módulo é responsável por agrupar os intervalos de 15 minutos de modo a obtermos a atividade realizada num dia de forma a ser possível utilizar o algoritmo de *Meyer*. Após a obtenção dos dias completos, agrupamo-los num conjunto de dias. Quando chegar a um número predefinido de dias (neste projeto estipulado em 28, porém adaptável), esse conjunto é utilizado para o cálculo de padrões de atividade. Tendo esse conjunto, ele é dividido em 2 *clusters*, onde em cada um deles estão inseridos os dias com características mais aproximadas. Foi tomada a decisão de ser utilizado dois *clusters* após análise de casos de testes. Verificou-se uma

distinção mais clara entre dias nos dois *clusters* do que com três, sendo possível ter uma maior certeza em que *cluster* os dias devem pertencer. O processo de *clustering* é feito através da plataforma R, através de um servidor R, onde são passados os valores em java da distância entre dias (neste caso 28) e através de bibliotecas R, esta faz o *clustering* e devolve à classe java o resultado. A decisão do *clustering* ser feito em R provém do facto que o R possui uma biblioteca (denominada “fpc”) que dada uma matriz de distância entre dias, faz o processo de *clustering* de forma dinâmica e automática, aspetos esses que não acontecem na linguagem java, onde a tarefa seria feita de forma ineficiente e sem recurso a bibliotecas semelhantes. Este processo de recolha de informação foi baseado a partir da ideia de Meyer (Meyer et al., 2016), sendo os dados usados para testar a implementação retirados do seu *dataset* (onde foram introduzidos na Tabela *StepCount*, a qual possui os dados da atividade física). A partir dessa divisão em 2 *clusters*, é criada numa tabela da base de dados (chamada *Patterns*) uma entrada por cada *cluster* existente, relacionada com a informação que se pode obter do *cluster* (por exemplo número de dias pertencentes ao *cluster*, média de passos a cada 15 minutos de intervalo num dia, desvio padrão dessa média, diferença de passos entre dia representativo do *cluster* e todos os dias pertencentes a este, entre outras informações). Esta informação será necessária para o módulo seguinte do *PIQL*. Este módulo possui um ficheiro de configuração que indica:

- O utilizador para o qual devem ser calculados os padrões.
- Quantos *clusters* devem ser calculados.
- A quantidade de intervalos que os dias podem ser atrasados/adiantados de forma a haver uma melhor comparação entre dias.
- Número de dias a serem calculados para os padrões de atividade.
- Número de dias para voltar a calcular os padrões.

DeviationPatternFinal é o segundo módulo. Este módulo está desenhado para pegar na informação recolhida nos *clusters* e definir não só o *cluster* em que o novo dia se situa, mas também se existiu um desvio de comportamento do utilizador relativamente ao seu padrão, e se esse desvio foi positivo ou negativo. A parte inicial do módulo é ler da tabela *Patterns* a informação de cada *cluster*. De seguida, calcula-se a distância entre os dados do novo dia e os dados dos *clusters*, onde o *cluster* que possui a menor distância, é considerado o *cluster* onde o novo dia se insere. Para saber se houve desvio e qual o seu tipo (positivo ou negativo), foram criadas cinco aproximações para cálculo de distância sendo o processo base referenciado na secção Detecção de desvios e as fórmulas e pseudo-código na secção 4.2.2.1 Pseudo-código e fórmulas das aproximações. A implementação

das aproximações assemelha-se à descrição dada nas secções descritas acima. Antes da execução de qualquer uma das aproximações, o primeiro passo foi ler e obter os valores do ficheiro de configuração associado a este módulo. Posteriormente foi necessário obter os dados do novo dia, seguido dos dados de ambos os *clusters*. Obtidos estes dados, através da fórmula de *Meyer*, calculou-se a distância entre o novo dia e ambos os *clusters*. A menor das distâncias indica qual o *cluster* a que o novo dia pertence. Após este processo é processado a aproximação desejada:

- Aproximação 1 (representado na Figura 16): Criado um método que recebe o valor de passos do novo dia, o número do *cluster* que o novo dia pertence, diferença de distância entre novo dia e dia representativo do seu *cluster* e valor de passos do dia representativo do *cluster*. O passo inicial é retirar da base de dados o valor de três variáveis necessárias para a aproximação funcionar (declaradas na secção 4.2.2.1 Pseudo-código e fórmulas das aproximações). Obtidos estes dados é usada a fórmula de *Meyer* para obter a distância entre o novo dia e o dia representativo e posteriormente usada a fórmula de desvio e de tipo de desvio referenciada na secção acima mencionada.

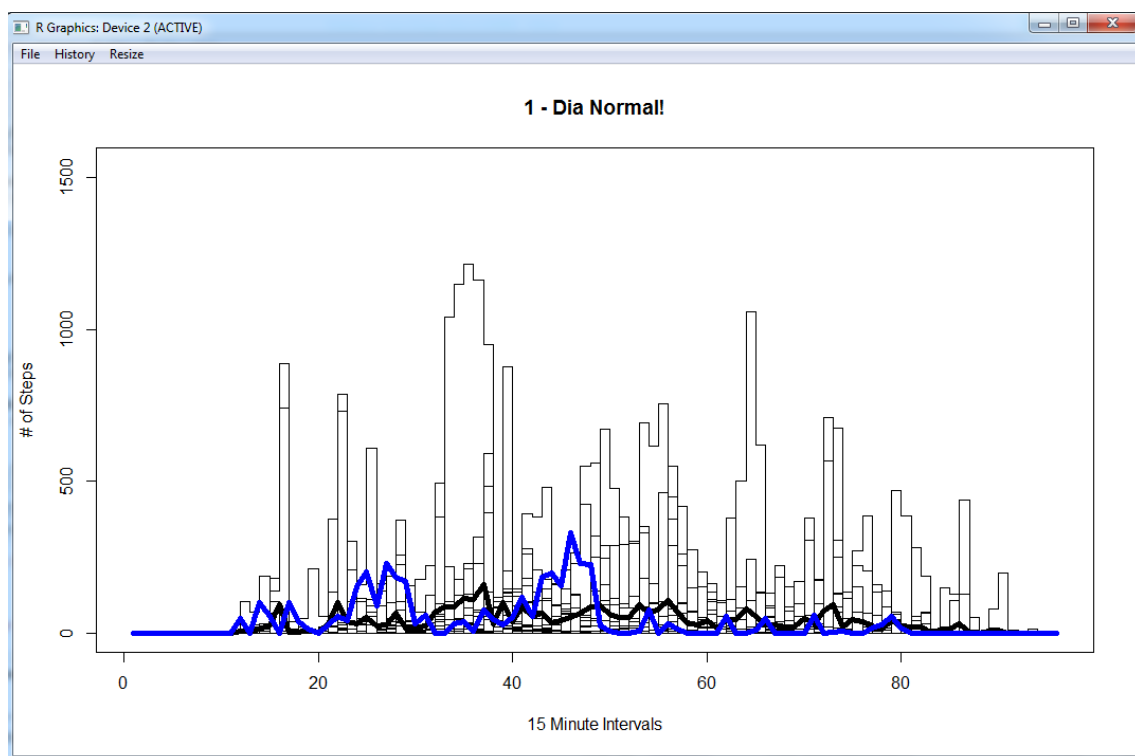


Figura 16. Representação gráfica da aproximação 1 (feita através da plataforma R).

- Aproximação 2 (representada na Figura 17): Antes de ser chamada a função relacionada com a aproximação dois, o primeiro passo é usar a fórmula de energia para transformar os passos do novo dia para energia, obtendo a energia do novo dia. De seguida chama-se a função da aproximação dois, a qual recebe o valor de energia do novo dia e o número do *cluster* que o novo dia pertence. Nessa função é inicialmente retirado da base de dados a energia gasta em média por dia em cada *cluster* e o desvio padrão dessa média para posteriormente utilizar-se a fórmula de desvio e do tipo de desvio referenciadas na secção 4.2.2.1 Pseudo-código e fórmulas das aproximações).

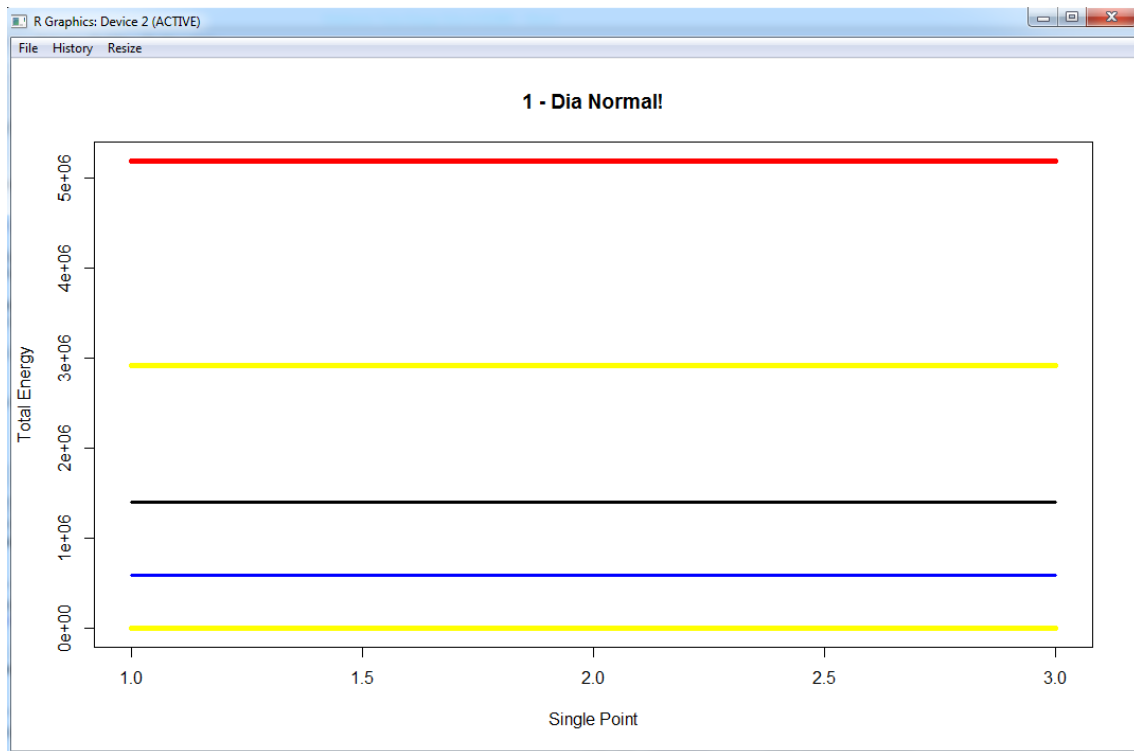


Figura 17. Imagem representativa da aproximação 2. Preto – Energia do dia representativo; Azul – Energia do Novo dia; Amarelo – Limite do desvio pequeno; Vermelho – Limite do desvio grande. Fornecido através da Plataforma R

- Aproximação 3 (observável na Figura 18): O passo inicial é chamar a função que processa a aproximação três. Esta recebe como parâmetros os passos do novo dia, o valor de passos de cada intervalo de 15 minutos do dia representativo do *cluster* em que se situa o novo dia e o desvio padrão calculado em cada intervalo. O passo seguinte é a criação de duas funções, que usando a fórmula descrita na secção 4.2.2.1 Pseudo-código e fórmulas das aproximações, que retornem os limites (inferior e superior) de passos para cada intervalo. Posteriormente foi criada duas funções, sendo que uma conta o número de limites superiores ultrapassados e a outra os valores que ficaram abaixo dos limites inferiores. Retornados estes dois valores e atendendo à informação já possuída, usa-se a fórmula de desvio e tipo de desvio descrita na secção acima mencionada.

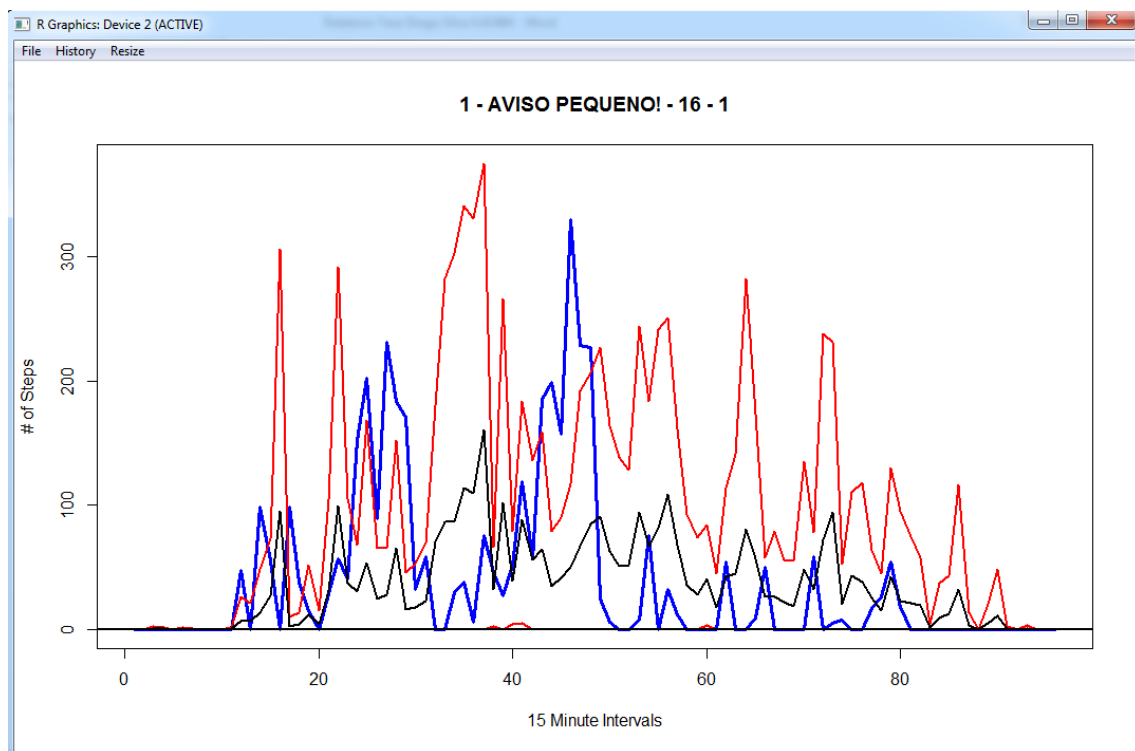
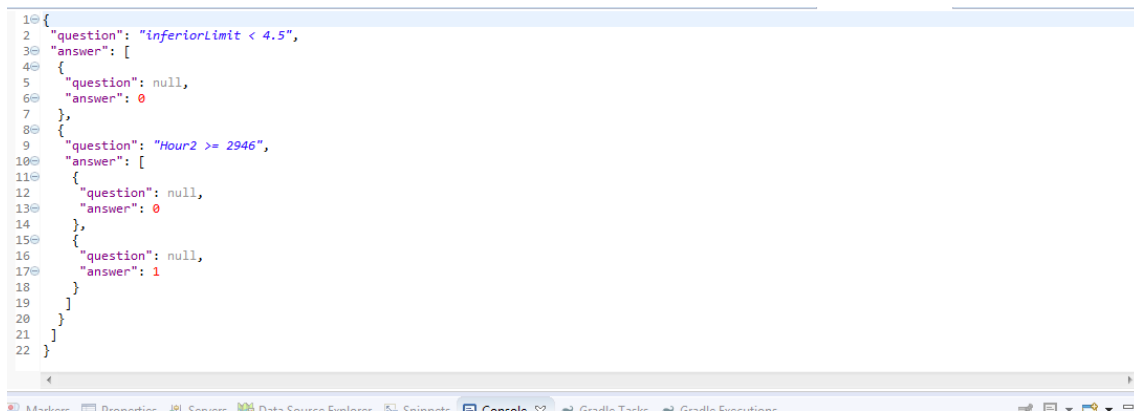


Figura 18. Imagem representativa da aproximação 3. Preto – Passos do dia representativo; Azul – Passos do Novo dia; Vermelho Superior – Limite superior; Vermelho Inferior – Limite Inferior. Fornecido através da Plataforma R.

- Aproximação 4: Aproximação semelhante à aproximação 3. O passo inicial é chamar a função que processa a aproximação quatro. Esta recebe como parâmetros os passos do novo dia, o valor de passos de cada intervalo de 15 minutos do dia representativo do *cluster* em que se situa o novo dia e o desvio padrão calculado em cada intervalo. O passo seguinte é a criação de duas funções, que usando a fórmula descrita na secção 4.2.2.1 Pseudo-código e fórmulas das aproximações, que retornem os limites (inferior e superior) de passos para cada intervalo. Posteriormente foi criada duas funções, sendo que uma conta o número de passos acima dos limites superiores e a outra os valores os números de passos abaixo dos limites inferiores. Retornados estes dois valores, cria-se uma variável que soma ambos os valores. De seguida cria-se uma variável que guarda o número total de passos dados no novo dia. Através destas últimas duas variáveis é possível calcular a percentagem de passos acima e abaixo dos limites. Obtendo esta variável, usa-se a fórmula de desvio e tipo de desvio descrita na secção acima mencionada.
- Aproximação 5: A aproximação é feita em dois passos. O primeiro passo passa pela recolha e criação das variáveis para a árvore de decisão e o segundo passo pela análise da árvore e obtenção do desvio. No primeiro passo, o primeiro processo passa pela chamada da função que recolhe e cria as variáveis da árvore. De seguida guarda-se em variáveis as informações que já existem na base de dados e que são necessárias para a árvore de decisão. Das variáveis necessárias (e descritas em 4.2.2.1 Pseudo-código e fórmulas das aproximações), as que são retiradas da base de dados são a distância entre o novo dia e o dia representativo do seu *cluster*, distância entre dia representativo do *cluster* do novo dia e o dia mais afastado, distância entre dias representativos dos *clusters* e energia média desperdiçada por dia no *cluster*. Exceto as variáveis relativas aos passos dados em certas partes do dia, as restantes variáveis descritas são calculadas da mesma forma que foram calculadas nas aproximações. É preciso voltar a recalculá-las uma vez que só se executa uma aproximação de cada vez, logo não é possível correr a aproximação 5 e obter variáveis que necessitamos e que foram calculadas por exemplo na aproximação 4. Para os passos dados entre certas horas do dia,

foi criada uma função que passado o valor de passos e um parâmetro de auxílio, dinamicamente soma os passos nesse intervalo. Para finalizar o primeiro passo, insere-se estas variáveis numa tabela da base de dados e inicia-se o segundo passo. Este inicia-se com a leitura de um ficheiro *JSON* que possui a árvore de decisão. O ficheiro (exemplificado na Figura 19), possui nele uma árvore de decisão já totalmente treinada, através de casos de teste, possuindo os dados necessários com o desvio.



```


1 {
2   "question": "inferiorLimit < 4.5",
3   "answer": [
4     {
5       "question": null,
6       "answer": 0
7     },
8     {
9       "question": "Hour2 >= 2946",
10      "answer": [
11        {
12          "question": null,
13          "answer": 0
14        },
15        {
16          "question": null,
17          "answer": 1
18        }
19      ]
20    }
21  ]
22 }

```

Figura 19. Exemplo de um ficheiro *JSON* com a árvore de decisão a ser analisada.

Criou-se uma função recursiva que tem como objetivo obter a questão e através da variável da questão, vai-se respondendo à pergunta, até obter-se o valor do desvio. No caso da Figura 19 verifica-se inicialmente se $\text{inferiorLimit} < 4.5$ (Se os intervalos que ficam abaixo do limite inferior da aproximação 3 são menos que 5). Será verificado na base de dados o valor da variável, e prosseguindo no exemplo, se for inferior a 4.5, vai para a questão “*null*”, indicando que não há mais perguntas, logo retorna-se o valor (answer: 0). Caso seja maior que 4.5, verifica-se a próxima pergunta, se $\text{Hour2} \geq 2946$ (se os passos dados entre as 6 e as 12 foram mais de 2946) e consoante o seu valor, vai-se descendo pela árvore até retornar um valor.

Independentemente da aproximação usada, no final é inserido na base de dados o tipo de desvio o qual será usado pelo módulo seguinte. Este módulo possui um ficheiro de configuração (visível na Figura 20) onde são armazenadas as constantes a serem utilizadas para a aproximação desejada. O valor destas constantes surgem de testes realizados com casos de testes realizados (explicados na secção 4.2.2) em que os melhores resultados obtidos foi com estes valores nas constantes.



```
1 user = 1
2 clusters = 2
3 approximationOption = 3
4 lag = 2
5 warningClasses = 2
6
7 Approximation 1
8 maxFactor = 1
9 sdFactor = 1
10
11 Approximation 2
12 a2constantBig = 2
13 a2constantLittle = 0.8
14
15 Approximation 3
16 a3constantUp = 0.8
17 a3constantDown = 0.7
18 lightRange = 10
19 seriousRange = 25
20
21 Approximation 4
22 a4constantUp = 0.9
23 a4constantDown = 0.5
24 a4PercentageLittle = 10
25 a4PercentageBig = 33
```

Figura 20. Ficheiro de Configuração do módulo *DeviationPatternFinals*. Possível observar as constantes para cada aproximação, juntamente com a aproximação a ser utilizada

COMBFinal é o terceiro módulo criado para o sistema *PIQL*. Este módulo é o módulo relacionado com o modelo de comportamento escolhido para o sistema *PIQL*. Este tem como objetivo, caso exista um desvio, averiguar através do modelo comportamental COMB os fatores que podem explicá-lo. Em caso do desvio ser nulo, o programa insere na base de dados que não há fatores que explicam o desvio. Caso haja um desvio, analisa-se o tipo de desvio (se positivo ou negativo) e recolhe-se todos os fatores que podem ter originado esse tipo de desvio (por exemplo se o tempo local está bom, pode explicar o desvio positivo). Desta lista de fatores, através da base de dados, são verificados os fatores que possuem parâmetros e valores associados a esses parâmetros (por exemplo será verificado se para o tempo local existe o parâmetro tempo juntamente com um link do *Context Manager*). Os valores dos parâmetros são dados preferencialmente através de *links* do *Context Manager*. Em termos de código esta validação dos fatores é feita através da leitura à base de dados, em que caso algum dos campos tanto do parâmetro como do seu valor estejam vazios, significa que não é possível retirar conclusões acerca do efeito desse fator no desvio do indivíduo. Caso exista fatores com múltiplos parâmetros e valores a eles associados, estes serão verificados. Se faltar algum parâmetro ou valor, embora todos os outros parâmetros do fator estejam completos, o fator é descartado. O módulo acaba com a impressão dos fatores iniciais selecionados juntamente com a inserção dos fatores e o número total destes na base de dados.

BCTSelectionFinal é o quarto módulo, encarregue de escolher as técnicas de persuasão a serem utilizadas (as quais denominamos de *BCT's*, derivado de *Behavior Change Techniques*). O módulo é iniciado com a obtenção dos fatores que podem explicar o desvio de comportamento. Com esta informação, é verificado se o endereço dos parâmetros associados a cada um desses fatores é válido. Caso existam parâmetros com endereços errados ou não-existent, o fator é descartado. Posteriormente, é analisado se

os endereços possuem valores que possam ser retirados e associados ao seu parâmetro. Caso não seja possível obter essa informação, o fator é descartado. Neste ponto possui-se a informação dos fatores que podem explicar o desvio, juntamente com os seus parâmetros e valores associados (de forma exemplificativa, possuímos para o fator tempo local o parâmetro tempo, com um endereço que indica o valor desse parâmetro).

Através destes dados, o passo seguinte é obter o fuso horário do utilizador. Esta informação situa-se guardada na base de dados. Com os fatores que explicam possivelmente o desvio e o fuso horário do utilizador, é verificado quais as regras que são aplicadas neste contexto, evidenciadas na Figura 21.

Adminer 4.1.0

DB: personal

Comando SQL Importar Exportar Criar tabela

registos BCTMessages
registos DecisionTree
registos Deviation
registos FactorAndBCTReturned
registos FactorsEffect
registos MessagesCreated
registos Pattern
registos RuleTable
registos StepCount
registos UserId
registos ValidFactors

Selecionar dados Mostrar estrutura Modificar estrutura Novo Registro

Selecionar Procurar Ordenar Limite 50 Tamanho do texto 100 Ação

Registo modificado, 21:08:32 Comando SQL

SELECT * FROM "RuleTable" LIMIT 50 (0,000 s) Modificar

	factor	file	parameters	bct	triggerHour	triggerDay
<input type="checkbox"/> modificar	Unfit	UnfitRule1.class	Steps HistorySteps	Positive Reinforcement Rational	17/18	
<input type="checkbox"/> modificar	Unfit	UnfitRule2.class	Steps HistorySteps	Positive Reinforcement Rational Social	17/18	
<input type="checkbox"/> modificar	Local weather	LocalWeatherRule1.class	Weather StepCount StepCountGoal	Social Positive Reinforcement	10	
<input type="checkbox"/> modificar	Local weather	LocalWeatherRule2.class	Weather StepCount StepCountGoal	Problem Solving	10	
<input type="checkbox"/> modificar	Fit	FitRule1.class	Fitness Steps HistorySteps	Rational	16/17/18	Sunday
<input type="checkbox"/> modificar	Fit	FitRule2.class	Fitness Steps HistorySteps	Rational Positive Reinforcement	16/17/18	

Figura 21. Lista de regras que explicam o possível desvio do utilizador.

Para cada fator está associado uma ou mais regras que podem explicar o desvio, os parâmetros necessários para cada regra ser avaliada, horas e dias a que esta será analisada. As regras estão inseridas no sistema através de uma pasta denominada “ruleFiles” a qual possui as regras no formato .class. O primeiro passo é através dos fatores que já possuímos, verificar quais as regras que podem vir a ser analisadas (Se obtivemos os fatores Fit e Local weather, as regras a serem analisadas são FitRule1, FitRule2, LocalWeatherRule1 e LocalWeatherRule2). Ao mesmo tempo é verificado o valor dos parâmetros associados a cada regra. Caso algum dos parâmetros não possua um valor, a regra é descartada. Obtendo estas regras, verifica-se as horas e dias a que estas devem ser despoletadas. A cada utilizador está associado um fuso horário, que irá permitir saber a hora atual e o dia do utilizador. Caso a hora e o dia não coincida com as horas e dias de ser acionada a regra, a regra é descartada (exemplificando, das quatro regras que possuímos, é analisada a hora corrente do utilizador e o seu resultado é 10. Como as regras de FitRule só são acionadas as 16, 17 e 18 horas, estas são descartadas). Obtendo as regras com fatores e parâmetros válidos e na hora desta serem acionadas, as regras são analisadas. A análise faz-se através de reflexão. Através do nome da classe a ser analisada, e dos seus parâmetros, é possível chamar a função. Este método permite proteger a

integridade do código presente na regra bem como agrupar as regras num local distinto dos módulos e ficheiros de configuração. Cada classe possui um método booleano que retorna verdadeiro caso uma condição relacionada com os parâmetros se verifique, caso contrário retorna falso, como é observável na Figura 22.

```
public class LocalWeatherRule1 {  
    public static boolean localWeatherRule1(String localWeather, double stepCount, double stepCountGoal) {  
        if(localWeather.equals("good") && stepCount < stepCountGoal)  
            return true;  
        else  
            return false;  
    }  
}
```

Figura 22. Regra *LocalWeatherRule1*. Retorna verdadeiro caso o tempo esteja bom e os passos dados estejam abaixo do objetivo. Caso contrário retorna falso. O método booleano deve possuir o mesmo nome que a regra e o caracter inicial minúsculo (*localWeatherRule1*).

É verificado quais as regras que retornaram verdadeiro, e através da base de dados, obtém-se as técnicas de persuasão a serem utilizadas (neste caso se *LocalWeatherRule1* retorna verdadeiro e *LocalWeatherRule2* retorna falso, a mensagem persuasiva a ser dada ao utilizador deve possuir uma vertente social e um reforço positivo). Os fatores e técnicas de persuasão a serem utilizadas são guardadas na base de dados para serem utilizadas pelo último módulo.

BCTInstanceFinal é o último módulo do sistema *PIQL*. O módulo recebe os fatores e técnicas persuasivas consideradas no módulo anterior e através das várias mensagens associadas aos fatores e técnicas persuasivas, aglomera-as e constrói uma mensagem de persuasão final. O primeiro passo é através dos fatores e técnicas persuasivas, obter a mensagem relacionada com estes parâmetros (como visível na Figura 23). Através do caso exemplificativo que tem sido abordado, foi retornado o fator Local Weather com as técnicas de persuasão social e reforço positivo. O módulo vai retornar a mensagem bem como a sua prioridade. A prioridade é uma variável criada para melhorar a estrutura da mensagem final, sendo que mensagens com prioridade 1 serão inseridas no início da mensagem final e consoante o nível de prioridade final aumentar, essa mensagem irá aparecer mais no final da mensagem persuasiva. Caso exista múltiplas mensagens para cada fator e técnica de persuasão (por exemplo na Figura 23, o tempo local com técnica de persuasão social possui 2 mensagens), é escolhida aleatoriamente uma das mensagens a ser selecionada para a mensagem final (de forma a diminuir a probabilidade de aparecer a mesma mensagem em dias consecutivos). Este processo será repetido para todos os fatores e técnicas persuasivas, resultando num conjunto de mensagens a serem inseridas para a mensagem inicial. As mensagens são posteriormente ordenadas através da prioridade, resultando na mensagem final. Obtendo a mensagem

final, é verificado se existe variáveis dentro da frase que seja necessário através do *Context Manager* obter o seu valor (como visível na Figura 23, a nomenclatura é <var>Caminho do *Context Manager* para obter a variável</var>). O último passo é feito pela criação de uma função que transforma a mensagem final num formato JSON, que vai permitir o envio desta para o *Adaptation Engine*.

SELECT * FROM 'BCTMessages' LIMIT 50 (0,000 s) Modificar

<input type="checkbox"/> Modify	factor	bct	message	priority
<input type="checkbox"/> modificar	Unfit	Social	What about invite someone to walk with you?	4
<input type="checkbox"/> modificar	Unfit	Social	Would you like to join a walking group?	4
<input type="checkbox"/> modificar	Unfit	Rational	You have reached <var>environment/stepPercentage</var>% of your goal	2
<input type="checkbox"/> modificar	Unfit	Rational	This week you have walked <var>steps</var> steps of your <var>stepGoals</var> steps goal	2
<input type="checkbox"/> modificar	Unfit	Positive Reinforcement	Congratulations	1
<input type="checkbox"/> modificar	Unfit	Positive Reinforcement	You have almost made it	1
<input type="checkbox"/> modificar	Unfit	Positive Reinforcement	This week you have earned <var>stepPoints</var> points	1
<input type="checkbox"/> modificar	Unfit	Problem Solving	Call Anne-Marthe	3
<input type="checkbox"/> modificar	Local Weather	Problem Solving	Call a cab and go to the mall	2
<input type="checkbox"/> modificar	Local Weather	Problem Solving	Do some activity at home	2
<input type="checkbox"/> modificar	Local Weather	Social	What about invite someone to walk with you?	3
<input type="checkbox"/> modificar	Local Weather	Social	Would you like to join a walking group?	3
<input type="checkbox"/> modificar	Local Weather	Positive Reinforcement	The weather is nice, go outside and earn 100 points	1
<input type="checkbox"/> modificar	Fit	Rational	You have reached <var>environment/stepPercentage</var>% of your goal	2
<input type="checkbox"/> modificar	Fit	Rational	This week you have walked <var>steps</var> steps of your <var>stepGoals</var> steps goal	2
<input type="checkbox"/> modificar	Fit	Positive Reinforcement	Congratulations	1
<input type="checkbox"/> modificar	Fit	Positive Reinforcement	You have almost made it	1
<input type="checkbox"/> modificar	Fit	Positive Reinforcement	This week you have earned <var>stepPoints</var> points	1

Figura 23. Tabela com as mensagens associadas a cada fator e técnica persuasiva.

5.3 Base de dados

Como referenciado na secção Ferramentas utilizadas e evidenciado na secção anterior, existe a necessidade de possuir uma base de dados para guardar estados e resultados gerados pelos módulos do sistema *PIQL*. A base de dados é em *MySQL* a qual é apoiada pela ferramenta *Adminer*. A escolha foi feita pelo facto de que *MySQL* é *open-source* de fácil utilização, pode ser usado em múltiplas plataformas, fornece segurança, possui um desenvolvimento rápido e com *updates*, sendo fácil de conciliar com Java, mantendo-se até este momento como uma das mais utilizadas¹⁵. A ferramenta escolhida foi o *Adminer*, pois é só necessário um ficheiro *PHP* de tamanho reduzido, suporta vários tipos de base de dados, bem como possui uma melhor performance comparativamente com ferramentas semelhantes¹⁶. Na Figura 24 é possível observar as diferentes tabelas da base de dados. Existem 11 tabelas de base de dados, cada uma com a sua função.

¹⁵ Rankings de base de dados relacionais, <https://db-engines.com/en/ranking/relational+dbms> , Visto em 4 de Julho de 2017.

¹⁶ Mais vantagens em <https://www.adminer.org/en/phpmyadmin/> , Visto em 4 de Julho de 2017

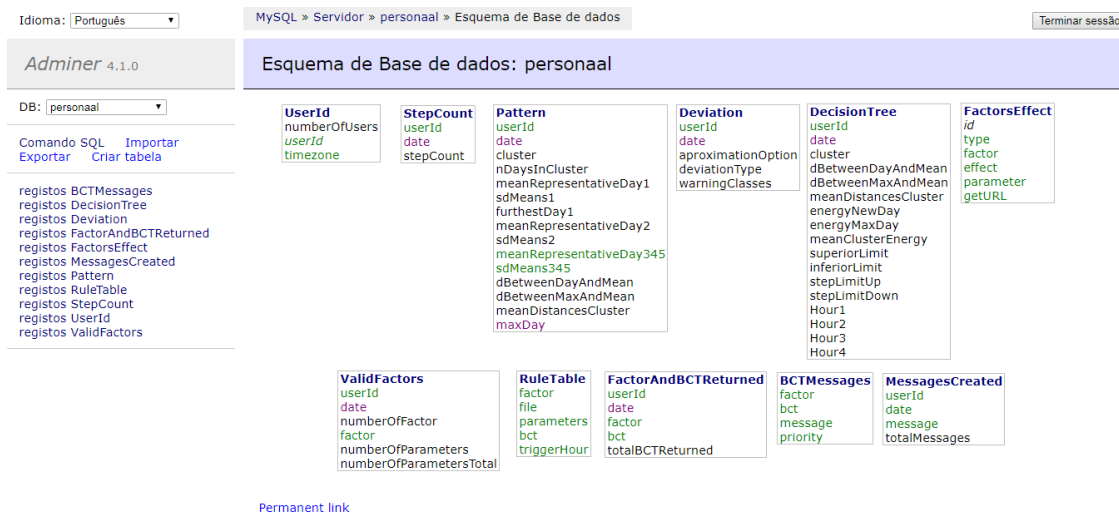


Figura 24. Esquema fornecido pelo Adminer relativo à base de dados. A verde estão variáveis do tipo “text”, a roxo “data”, as restantes são inteiros ou strings

A tabela *UserId* foi criada com o intuito de saber os utilizadores pertencentes à aplicação. Possui uma variável de incremento automático (*numberOfUsers*), juntamente com o utilizador e o seu fuso horário. Fundamental para o reconhecimento dos utilizadores nos módulos do sistema PIQL.

A tabela *StepCount* é a tabela usada no módulo *IdentifyPatternsFinal* e *DeviationPatternsFinal*. Cada entrada desta tabela possui o nome do utilizador, a data de cada intervalo de 15 minutos (no formato 2017-10-07 00:15:00) juntamente com os passos dados nesse intervalo. Guarda a informação necessária para calcular os padrões de atividade bem como os valores do novo dia.

A tabela *Pattern* guarda os valores obtidos da execução do módulo *IdentifyPatternsFinal*. Cada entrada representa os dados obtidos de um *cluster* de um determinado utilizador. A tabela é utilizada no módulo *DeviationPatternsFinal* com o intuito de obter os dados necessários dos padrões de atividade para calcular o tipo de desvio do utilizador.

A tabela *Deviation* tem como finalidade registar o tipo de desvio calculado por uma aproximação. Esta é preenchida no final da execução do *DeviationPatternFinal* com o id do utilizador, a data de preenchimento do registo, a aproximação que foi usada, o tipo de desvio e a classe de desvio.

A tabela *DecisionTree* é necessária para guardar as variáveis necessárias para a árvore de decisão gerada no módulo *DeviationPatternsFinal*. A explicação das variáveis estão na secção 4.2.2.1 Pseudo-código e fórmulas das aproximações, na aproximação 5.

A tabela *FactorsEffect* é a tabela que possui a lista de fatores baseada no modelo comportamental COMB. Esta possui em cada registo uma variável de incremento, o tipo de fator (se é fator de capacidade, oportunidade ou motivação, COM), o fator, o efeito que o fator tem no tipo de comportamento do utilizador, o parâmetro associado ao fator e o endereço do *Context Manager* em que pode ser obtido o valor desse parâmetro. Para fatores com múltiplos parâmetros, são criadas múltiplas entradas. A tabela é usada no módulo *COMBFinal*, com o objetivo de descobrir os possíveis fatores que explicam o desvio analisado no módulo *DeviatonPatternsFinal*.

A tabela *ValidFactors* guarda os possíveis fatores que explicam o desvio, gerados no módulo *COMBFinal*. Cada registo possui o id do utilizador, a data que foi gerado o registo, o número do fator (gerado automaticamente através de uma variável incrementável), o fator, número de parâmetros associados a esse fator e o número de parâmetros registados no final. A informação é depois retirada para ser usada no módulo *BCTSelectionFinal*.

A tabela *RuleTable* possui as regras que explicam desvios por parte do utilizador. Cada entrada possui um fator, o ficheiro com a regra relacionado com esse fator, os parâmetros necessários para analisar a regra, as técnicas de persuasão que retorna caso a regra seja confirmada e as horas e dia a que a regra deve ser ativada. Cada entrada é para cada ficheiro de regra, sendo que um fator pode ter múltiplos ficheiros de regras. A informação inserida nesta tabela é usada no módulo *BCTSelectionFinal*, em que para cada fator que explica o desvio, será verificada se possui regras associadas. Caso exista, será fornecido o nome do ficheiro da regra e após execução da mesma no *BCTSelectionFinal*, caso retorne verdade, a base de dados retorna as técnicas de persuasão associadas ao fator e à regra.

A tabela *FactorAndBCTReturned* é usada para registar os valores após a execução do módulo *BCTSelectionFinal*. Para cada entrada é guardado o id do utilizador, a data de registo de cada entrada, o fator, a técnica de persuasão associada a esse fator e o número de *BCT's* retornados. Para cada técnica de persuasão associada é criada uma entrada, sendo incrementado o número de *BCT's* finais. É retirada da tabela a informação no decorrer do último módulo.

A tabela *BCTMessages* possui uma lista de mensagens referentes a cada fator e *BCT* associado, juntamente com uma prioridade que ajuda na construção da mensagem final. Se para um dado fator e técnica de persuasão a si associado, existirem múltiplas mensagens, é criada uma entrada para cada mensagem. A tabela é chamada pelo módulo *BCTInstanceFinal* de forma a retornar as mensagens associadas bem como a prioridade das mesmas.

A tabela *MessagesCreated*, última tabela da base de dados, é desenhada para guardar as mensagens que são enviadas para o *AdaptationEngine*. Após ser confirmado o envio da mensagem com sucesso, é guardado na base de dados o id do utilizador, a data de registo da mensagem, a mensagem final e o número de mensagens utilizadas para originar essa mensagem.

5.4 Interface *REST*

O sistema possui uma interface *REST*, sobre a qual podem ser feitos pedidos *POST*. Esta decisão provém da ideia do sistema *PIQL* ser flexível o suficiente para ser acionado por outros módulos do *PersonAAL* ou de qualquer aplicação que se ligue ao *PersonAAL* e necessite de outra forma de comunicação com o *PIQL*. A interface *REST* (visível na Figura 25) consiste na criação de três módulos, denominados *servlets*, que são responsáveis por receberem os pedidos *POST* feitos para a *API REST* do sistema *PIQL*. Os pedidos são realizados através de um endereço *URL* através de um texto com estrutura *JSON*. Os *servlets* verificam a sintaxe dos pedidos e caso esta esteja errada, o pedido é rejeitado, indicando os campos que estão errados. Caso a sintaxe esteja correta, o pedido é encaminhado para o módulo correspondente ao *servlet*, onde no final pode ou não originar uma mensagem persuasiva. O sistema *PIQL* possui 3 *servlets*, M3, M4 e M5. O pedido a M3 é posteriormente enviado para o *COMBFinal*, o pedido M4 dirigido para o *BCTSelectionFinal* e o M5 enviado para *BCTInstanceFinal*. Os pedidos a cada endereço são diferentes entre si, cada um com certos atributos. O pedido (o qual é feito através de *JSON*) pode estar mal formatado, nesse caso é informado ao utilizador que o seu pedido não foi processado. Caso o pedido esteja bem formatado, é analisado o seu conteúdo. Analisado o seu conteúdo e verificado a validade do pedido, pode ser possível ou não enviar uma mensagem persuasiva ao utilizador, dependendo dessa mesma verificação e das opções tomadas em cada pedido.

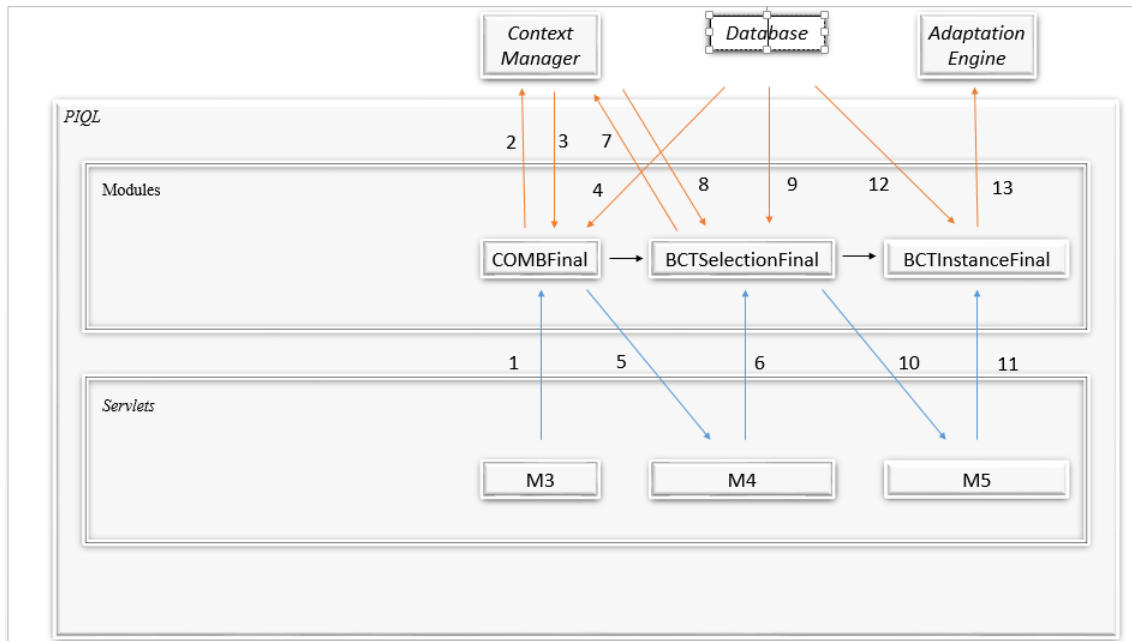


Figura 25. Esquema representativo do sistema PIQL através da interface REST. A numeração representa os passos de execução do sistema.

O primeiro *servlet* (M3) recebe pedidos através do URL “accessible-serv.lasige.di.fc.ul.pt:8081/m3”, que caso tenha uma sintaxe válida, é dirigido o pedido para *COMBFinal*. Um exemplo de pedido a ser feito ao primeiro *servlet* é:

```
{ "User": "1", "Date": "2017-07-02 19:56:11", "Approximation Option": "5", "Deviation Result": "2", "Warning Class": "2", "Run Option": "Global" }
```

Em que:

User é o utilizador sobre o qual irá incidir a regra. Este tem que pertencer à tabela *UserId* da base de dados. Caso o valor do parâmetro seja “All”, o pedido será realizado para todos os utilizadores que pertencem à tabela da base de dados. Caso contrario, o pedido será recusado.

Deviation Result é um inteiro que diz o tipo de desvio que ocorreu no utilizador. Em caso de o número ser positivo, significa que houve um desvio positivo em termos de comportamento do utilizador, caso seja negativo, houve um desvio negativo. Este inteiro tem que estar entre o valor do *Warning Class* e o módulo do valor do *Warning Class* (caso o *Warning Class* seja 2, o valor do *Deviation Result* deve situar-se entre o 2 e o -2). Caso o valor do *Deviation Result* seja 0, considera-se que não houve desvio, portanto o pedido é recusado, pois sem desvio não há necessidade de criar mensagens persuasivas.

Aproximation Option é também um inteiro que representa a aproximação sobre a qual o desvio foi calculado. Este valor tem que ser um inteiro entre 1 e 5 (número de aproximações existentes). Este atributo correntemente não é utilizado porém poderá vir a ser útil no futuro.

Date é a data atual do pedido. Tem que ter formato Ano-Mês-Dia Hora:Minuto:Segundo. Qualquer outro formato será recusado. Este atributo não é utilizado porém pode tornar-se relevante no futuro.

Warning Class é um inteiro que representa o número de classes de desvio existentes. Deve ter valor positivo. Este atributo correntemente não é utilizado porém poderá ser útil para estudos futuros.

Run Option é o parâmetro que indica como o processo deve ser executado. Este possui duas opções. A primeira opção, “Module”, permite ao utilizador executar o pedido *POST* e após ser executado o módulo *COMBFinal*, o pedido é finalizado. Caso seja usada a segunda opção, “Global”, o pedido será processado pelo módulo *COMBFinal* e posteriormente enviado o resultado dessa execução em formato de pedido *POST* para M4, e assim sucessivamente até ao final de execução do sistema *PIQL* (envio da mensagem persuasiva para o *Adaptation Engine*). Outro valor utilizado neste parâmetro é recusado.

Se a sintaxe estiver correta (cada parâmetro ser do tipo definido pela norma, não existirem parâmetros extras ou falta de parâmetros essenciais), o pedido é passado para o módulo *COMBFinal*, em que caso os valores estejam corretos, são retirados os valores do “User”, “Deviation Result” e “Run Option” e o pedido é processado. Caso não haja erros no processamento do pedido, o módulo cria um pedido *POST* para o M4 com os valores processados se o valor de “Run Option” seja Global, caso contrário o pedido é processado no *COMBFinal* e deixa de ser executado o sistema *PIQL*.

O M4 (*segundo servlet*) recebe pedidos através do URL “accessible-serv.lasige.di.fc.ul.pt:8081/m4”, que caso tenha uma sintaxe válida, é dirigido o pedido para *BCTSelectionFinal*. Um exemplo de pedido a ser feito ao segundo *servlet* é:

```
{ "User": "1", "Factors": "Fit/test", "Date": "2017-07-09  
19:05:00", "NumberOfFactors": 2, "Run Option": "Global" }
```

Em que:

User é o utilizador sobre o qual irá incidir a regra. Este tem que pertencer à tabela *UserId* da base de dados. Caso o valor do parâmetro seja “All”, o pedido será realizado

para todos os utilizadores que pertencem à tabela da base de dados. Caso contrário, o pedido será recusado.

Factors são fatores válidos que explicam o desvio. Podem ser consultados na tabela *FactorsEffect* da base de dados. Caso haja um fator inválido, o pedido será recusado. Caso existam fatores contraditórios (por exemplo um fator positivo e um negativo) o pedido também é recusado, pois se houve um desvio positivo, não interessa analisar os fatores negativos, o mesmo se aplica para o desvio negativo e fatores positivos. Caso o número de fatores representados no pedido não sejam o mesmo número que está representado em *NumberOfFactors*, o pedido é também ele recusado. Os diferentes fatores são separados por “/”. Caso não estejam separados por “/”, estes são considerados 1 fator apenas.

Date é a data atual do pedido. Tem que ter formato Ano-Mês-Dia Hora:Minuto:Segundo. Outro formato será recusado.

NumberOfFactors é o número de fatores válidos que são passados no atributo “Factors”.

Run Option é o parâmetro que indica como o processo deve ser executado. Este possui duas opções. A primeira opção, “Module”, permite ao utilizador executar o pedido *POST* e após ser executado o módulo *BCTSelectionFinal*, o pedido é finalizado. Caso seja usada a segunda opção, “Global”, o pedido será processado pelo módulo *BCTSelectionFinal* e posteriormente enviado o resultado dessa execução em formato de pedido *POST* para M5, e assim sucessivamente até ao final de execução do sistema *PIQL* (envio da mensagem persuasiva para o *Adaptation Engine*). Outro valor utilizado neste parâmetro é recusado.

Caso o pedido não possua a sintaxe correta (cada parâmetro ter um tipo diferente do estipulado ou existirem parâmetros diferentes dos definidos), o utilizador é informado acerca do insucesso do pedido. Caso este tenha boa sintaxe (cada parâmetro ser do tipo definido pela norma e não existirem outros parâmetros adicionais ou parâmetros em falta) é reencaminhado para o *BCTSelectionFinal*, onde se tiver os valores válidos são retirados os valores dos campos “User”, “Factors”, “NumberOfFactors” e “Run Option”. De seguida o *BCTSelectionFinal* processa o pedido. Caso existam erros no processamento do pedido, o programa pára e alerta que o pedido não foi bem-sucedido. Caso não existam erros no processamento do pedido, é feito um pedido *POST* ao M5 com os valores resultantes do processamento no *BCTSelectionFinal* se a opção “Run Option” tiver como valor “Global”, caso contrário o módulo é processado e deixa de ser executado o sistema *PIQL*.

O M5 (terceiro e último *servlet*) recebe pedidos através do URL “accessible-serv.lasige.di.fc.ul.pt:8081/m5” em que caso tenha uma sintaxe válida, é dirigido o pedido para *BCTSelectionFinal*. Um exemplo de pedido a ser feito ao último *servlet* é:

```
{"User":"john","Date":"2017-07-10 01:38:38.0","FactorAndBCT":"Fit - Positive Reinforcement/Unfit - Positive Reinforcement","TotalNumber":2}
```

Em que:

User é o utilizador sobre o qual irá incidir a regra. Este tem que pertencer à tabela *UserId* da base de dados, pois caso não pertença, o pedido será recusado.

Date é a data atual do pedido. Tem que ter formato Ano-Mês-Dia Hora:Minuto:Segundo. Outro formato será recusado.

FactorAndBCT tem os fatores e *BCT's* associados. Estes têm que ser válidos, e não podem ser contraditórios. Caso isto aconteça, o pedido será recusado. Separa-se fator e *BCT* por “ – “, sendo que cada fator e *BCT* associado separam-se dos restantes por “/”. Por exemplo temos fator “Fit” com “Positive Reinforcement” como *BCT*, e temos fator “Fit” com *BCT* “Rational”. O parâmetro a ser passado no pedido *POST* seria “FactorAndBCT”: “Fit - Positive Reinforcement/Fit - Rational”. Uma lista de fatores e *BCT's* associados aos mesmos podem ser encontrados na tabela *RuleTable* da base de dados.

TotalNumber é o número de *FactorAndBCT* válidos que são passados no atributo “FactorAndBCT”.

Novamente caso o pedido feito a M5 não possua a sintaxe correta o utilizador é informado acerca do insucesso do pedido. Caso este tenha boa sintaxe, será reencaminhado para o *BCTInstanceFinal*, onde se tiver valores válidos, são retirados os valores do campo “User”, e “FactorAndBCT”. De seguida o *BCTInstanceFinal* processa o pedido. Caso existam erros no processamento do pedido, o programa pára e alerta que o pedido não foi bem-sucedido. Caso não se detetem erros no processamento do pedido, é criado o ficheiro *JSON* a ser enviado para o *Adaptation Engine* com a mensagem persuasiva processada no módulo *BCTInstanceFinal*.

5.5 Servidor

O servidor guarda o código necessário para o funcionamento do sistema *PIQL* e a base de dados associada à mesma. A ponte entre o ambiente de desenvolvimento e o servidor, como referido na secção Ferramentas utilizadas, é feita através de um repositório GIT. O repositório (visível na Figura 26) possibilita a inserção de ficheiros que serão puxados para o servidor, para que após um pedido no servidor, esses ficheiros sejam inseridos.

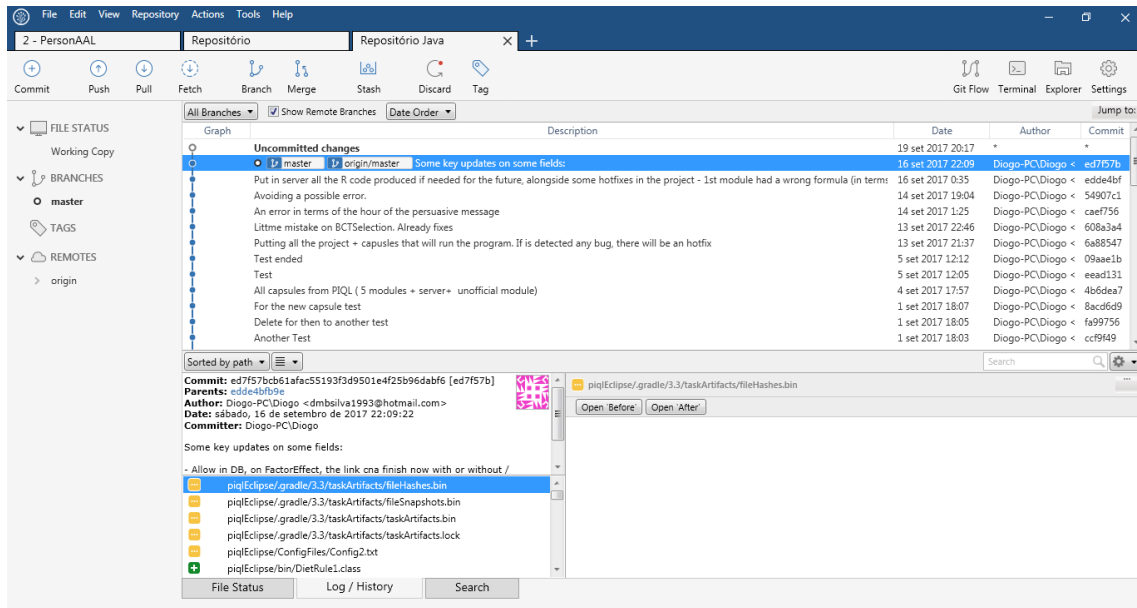


Figura 26. SourceTree, que possui o repositório que faz a transferência de ficheiros do ambiente de desenvolvimento para o servidor do *PIQL*.

5.5.1 Ficheiros inseridos no servidor

Os ficheiros inseridos no servidor são os indicados na Figura 27. Os ficheiros necessários para o pleno funcionamento do *PIQL* são 8 capsulas jar e 3 pastas.

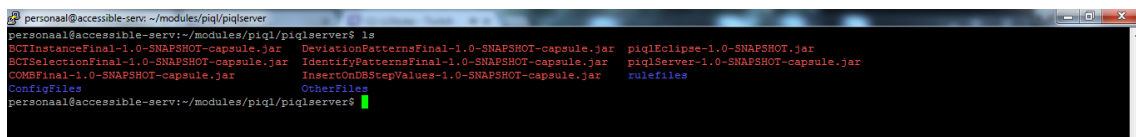


Figura 27. Visualização dos ficheiros necessários para a execução do sistema *PIQL*

Relativamente às 3 pastas, cada uma tem um propósito diferente. Na pasta *ruleFiles* estão inseridos todos os ficheiros de regras (.class) enunciados na tabela *RuleTable* e que verificam as condições que explicam o desvio de comportamento de utilizador. Na pasta *ConfigFiles* estão situados os ficheiros txt relacionado com os ficheiros de configuração de cada módulo (incluindo o ficheiro de configuração do módulo

insertOnDBStepValues). A pasta *OtherFiles* possui a *database* de *Meyer* juntamente com um ficheiro de configuração relacionado com a base de dados.

As cápsulas são originadas a partir do projeto *Gradle* inserido no *Eclipse*. É possível criar cápsulas que permitem correr um módulo como uma classe java sem haver a necessidade de preocupar com as dependências. Os passos para a criação de cápsulas estão indicados no Apêndice D – Criação de cápsulas. Cada cápsula possui um objetivo (como indicado na Tabela 9).

Nome da Cápsula	Objetivo
piqlEclipse-1.0-SNAPSHOT-capsule.jar	Cápsula criada automaticamente quando são criadas cápsulas num projeto GRADLE
InsertOnDBStepValues-1.0-SNAPSHOT-capsule.jar	Cápsula relacionada com o módulo insertOnDBStepValues que permite facilmente inserir atividade física do utilizador na tabela <i>StepCount</i>
IdentifyPatternsFinal-1.0-SNAPSHOT-capsule.jar	Cápsula do módulo <i>IdentifyPatternsFinal</i> (identificação de padrões)
DeviationPatternsFinal-1.0-SNAPSHOT-capsule.jar	Cápsula do módulo <i>DeviationPatternsFinal</i> (detecção de desvios)
COMBFinal-1.0-SNAPSHOT-capsule.jar	Cápsula do módulo <i>COMBFinal</i> (Modelação de comportamento)
BCTSelectionFinal-1.0-SNAPSHOT-capsule.jar	Cápsula do módulo <i>BCTSelectionFinal</i> (Seleção de técnicas de alteração de comportamento)
BCTInstanceFinal-1.0-SNAPSHOT-capsule.jar	Cápsula do módulo <i>BCTInstanceFinal</i> (Implementação de técnicas de alteração de comportamento)
piqlServer-1.0-SNAPSHOT-capsule.jar	Cápsula relacionada com a interface <i>REST</i> e servidor <i>jetty</i>

Tabela 9 - Cápsulas do sistema e seus objetivos

A cápsula `piqlServer-1.0-SNAPSHOT-capsule.jar` é criada a partir de uma classe java denominada `Main.java`. A necessidade desta classe passa pelo facto de como o servidor é Apache HTTP, este não permite processar os pedidos feitos aos *servlets* criados em java. Tendo conhecimento desta situação, a decisão tomada foi a construção de uma cápsula através de uma classe que possuísse um servidor *jetty* juntamente com o código dos *servlets*. Quando é feito um pedido aos *servlets* existentes no servidor Apache, o pedido é redireccionado para o servidor *jetty* (através de configurações no proxy) onde o pedido é depois processado.

5.6 Execução do sistema

A execução do sistema PIQL é feita ao nível do servidor, através das cápsulas. Existem duas opções distintas para executar o sistema (registadas na Tabela 10).

Opções de execução do sistema <i>PIQL</i>	
Através da linha de comandos (denominada por <i>CommandLine</i>)	Através da interface REST existente (denominada por <i>REST</i>)

Tabela 10 – Opções de execução do sistema *PIQL*

Através da interface *REST*, o sistema corre através de um pedido feito a um dos endereços indicados na secção Interface *REST*. Consoante o valor da opção “*Run Option*”, o sistema pode correr um módulo apenas (se o valor for “*Module*”) ou desde um módulo até à obtenção da mensagem persuasiva (caso o valor seja “*Global*”). Em termos de servidor, caso um pedido seja feito para `accessible-serv.lasige.di.fc.ul.pt:8081/m3` (primeiro *servlet*, M3), assumindo que todos os valores estão corretos e a opção do “*Run Option*” é “*Global*”, o pedido será encaminhado para o servidor *jetty*, o qual foi iniciado no servidor através da cápsula `piqlServer-1.0-SNAPSHOT-capsule.jar`. O pedido quando for recebido no servidor *jetty*, será processado no M3 a sua sintaxe, que caso esteja correta, chama o módulo *COMBFinal* para verificar os fatores que explicam o desvio sugerido pelo pedido POST feito. Caso existam fatores que expliquem o desvio, é feito no módulo um pedido POST para M4, no formato indicado na secção Interface *REST*, com os valores desses parâmetros dados pela execução do módulo. Em M4 será avaliada a sintaxe do pedido, processado posteriormente pelo módulo *BCTSelectionFinal*, enviado um pedido para M5, encaminhado para *BCTInstanceFinal* e concluído com a criação da mensagem persuasiva a ser enviada para o *AdaptationEngine*.

Através da linha de comandos, como o próprio nome indica, é necessário correr as cápsulas juntamente com argumentos associados à mesma. A Tabela 11 indica as opções que podem ser utilizadas.

Argumentos para a execução da opção CommandLine				
Comando universal	Nome da Cápsula	Opção CommandLine	Tipo de execução pretendida	Utilizador pretendido
java -jar	<Nome da cápsula>	<CommandLine>	<Module> ou <Global>	<vazio> ou <All>

Tabela 11 – Argumentos para a execução da opção CommandLine

Qualquer execução da cápsula inicia-se com o comando java -jar, de modo a conseguir extrair a informação que a cápsula contém. De seguida no comando é necessário indicar o nome da cápsula adicionalmente com a opção <CommandLine>. O argumento seguinte indica o tipo de execução pretendida, se é Module (se é só necessário correr o módulo inserido nessa cápsula) ou Global (necessário correr os módulos inteiros até se obter a mensagem persuasiva). O último argumento é o utilizador pretendido. Caso este argumento esteja vazio, indica que o comando deve ser aplicado ao utilizador que está inserido no ficheiro de configuração do módulo. Caso seja indicado <All>, o programa vai executar esse módulo para todos os utilizadores existentes na base de dados.

Em termos exemplificativos, o comando “java -jar IdentifyPatternsFinal-1.0-SNAPSHOT-capsule.jar CommandLine Module All” vai identificar os padrões de atividade (1º módulo) de todos os utilizadores da base de dados, sendo que após a execução desse módulo, o programa pára. O comando “java -jar COMBFinal-1.0-SNAPSHOT-capsule.jar CommandLine Global” vai analisar os fatores que podem explicar o desvio do utilizador que está inserido no ficheiro de configuração do módulo *COMBFinal* e vai executar os módulos seguintes até à inserção da sua mensagem persuasiva na base de dados. O primeiro comando pode ser visualizado na Figura 28 e o segundo na Figura 29, Figura 30 e Figura 31.

```
personaal@accessible-serv: ~/modules/piql/piqlserver
personaal@accessible-serv:~/modules/piql/piqlserver$ java -jar IdentifyPatternsFinal-1.0-SNAPSHOT-capsule.jar CommandLine Module All
Not a day to recalculate cluster.

Not a day to recalculate cluster.

Not a day to recalculate cluster.

IdentifyPatternsFinal Done. CommandLine Module Option
personaal@accessible-serv:~/modules/piql/piqlserver$
```

Figura 28. Visualização do comando de exemplo 1, desenhado para calcular o padrão de atividade dos utilizadores na base de dados. Existe 3 outputs dado que a base de dados possui 3 utilizadores. Após execução, devido à opção Module, o programa sai.

```
personaal@accessible-serv: ~/modules/piql/piqlserver
personaal@accessible-serv:~/modules/piql/piqlserver$ java -jar COMBFinal-1.0-SNAPSHOT-capsule.jar CommandLine Global
Factors Selected for user Terrence are:
Fit
Happiness
Activity Knowledge
Normal Grip Strength
Self-efficacy
Walking aids
Normal BMI
Activity Nearby Home
Affordable activity
Company (practical support)
Got equipment needed for the activity
Local safety
Local weather
Free Agenda
Schedule Activity
Over step goal
Company (social)
Impress caretaker
Interested in the activity
Introjected regulation (feelings of guilt or shame)
Perceived physical benefits
Perceived psychological benefits (well-being)
Perceived social benefits
Positive feelings and memories about the activity
test
test
Fit
Fit
Diet
Diet
Diet
Diet

Factors Selected are:
Fit
test
Diet

Starting factors for user Terrence are:
```

Figura 29. Execução do 2º comando de exemplo. Nesta parte é possível ver a análise dos fatores que podem explicar o desvio de comportamento do Terrence (utilizador no ficheiro de configuração)

```

personal@accessible-serv: ~/modules/piql/piqlserver
Starting factors for user Terrence are:
Fit
test
Diet

Factors with parameters which have valid URLs are:
test
Diet

Classes which have the valid parameters:
TestRule2.class
TestRule1.class

Classes triggered:
TestRule2.class
TestRule1.class

Factors and Bots Returned:
test - Rational
test - Positive Reinforcement

BCT's returned for user Terrence :
Positive Reinforcement
Rational

BCTSelection Done

Starting Messages for user Terrence are:
test - Positive Reinforcement - You really did it|
test - Rational - Impressive|

After Merge Messages:
Positive Reinforcement - You really did it|
Rational - Impressive|

After the draw, the messages are:
test - Positive Reinforcement - You really did it
test - Rational - Impressive

The user Terrence has a new Alert Messages!

```

Figura 30. Continuação do output obtido da execução do 2º comando do exemplo. Possível observar as regras que possuem parâmetros válidos, juntamente com as que foram despoletadas, BCTs que as regras retornam e aviso de mensagem persuasiva.

```

personal@accessible-serv: ~/modules/piql/piqlserver

Classes which have the valid parameters:
TestRule2.class
TestRule1.class

Classes triggered:
TestRule2.class
TestRule1.class

Factors and Bots Returned:
test - Rational
test - Positive Reinforcement

BCT's returned for user Terrence :
Positive Reinforcement
Rational

BCTSelection Done

Starting Messages for user Terrence are:
test - Positive Reinforcement - You really did it|
test - Rational - Impressive|

After Merge Messages:
Positive Reinforcement - You really did it|
Rational - Impressive|

After the draw, the messages are:
test - Positive Reinforcement - You really did it
test - Rational - Impressive

The user Terrence has a new Alert Messages!

Message is: Impressive. You really did it.
Making request to Adaptation Engine...
Information send successfully

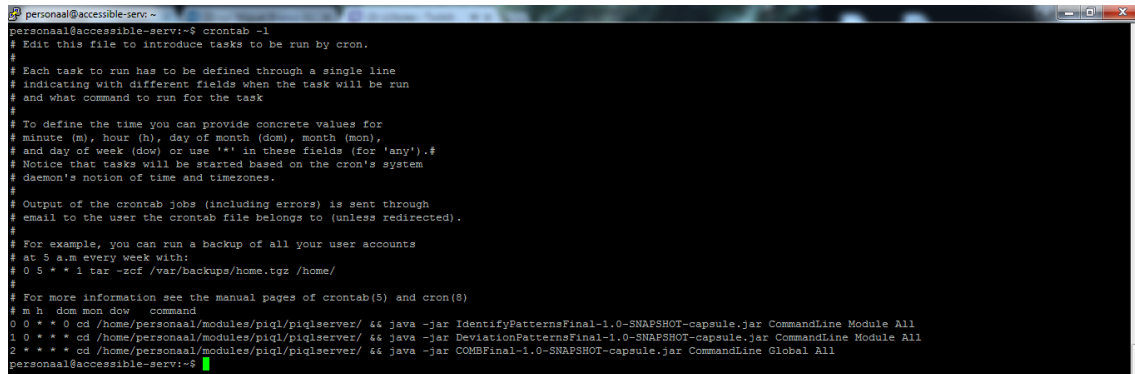
End of Program
personal@accessible-serv:~/modules/piql/piqlserver$

```

Figura 31. Parte final do output obtido da execução do 2º comando do exemplo. Possível observar aspectos da figura anterior. Possível observar a mensagem persuasiva obtida e a aceitação da mensagem por parte do Adaptation Engine.

O sistema foi desenhado também para correr automaticamente, através de crontab. Crontab permite que num servidor seja possível executar comandos a uma dada hora definida pelo utilizador. No caso do sistema *PIQL* foram introduzidos três comandos (observar Figura 32). O primeiro comando é para o cálculo dos padrões de atividade dos utilizadores serem feitos de forma semanal. O segundo comando permite diariamente

detetar desvios no padrão, caso existam. O terceiro e último comando possibilita analisar de hora a hora os fatores que podem explicar o desvio, as regras associadas a esses fatores e a criação da mensagem persuasiva a ser enviada para o utilizador.

A screenshot of a terminal window titled 'personaal@accessible-serv: ~'. The user has entered the command 'crontab -l'. The output shows the contents of the crontab file, which includes instructions on how to define tasks, an example of a backup task, and a list of scheduled tasks. The tasks are: 0 0 * * 0 cd /home/personaal/modules/piql/piqlserver/ && java -jar IdentifyPatternsFinal-1.0-SNAPSHOT-capsule.jar CommandLine Module All; 1 0 * * * cd /home/personaal/modules/piql/piqlserver/ && java -jar DeviationPatternsFinal-1.0-SNAPSHOT-capsule.jar CommandLine Module All; 2 * * * cd /home/personaal/modules/piql/piqlserver/ && java -jar COMBFinal-1.0-SNAPSHOT-capsule.jar CommandLine Global All.

```
personaal@accessible-serv:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dpm), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
m h dom mon dow   command
0 0 * * 0 cd /home/personaal/modules/piql/piqlserver/ && java -jar IdentifyPatternsFinal-1.0-SNAPSHOT-capsule.jar CommandLine Module All
1 0 * * * cd /home/personaal/modules/piql/piqlserver/ && java -jar DeviationPatternsFinal-1.0-SNAPSHOT-capsule.jar CommandLine Module All
2 * * * cd /home/personaal/modules/piql/piqlserver/ && java -jar COMBFinal-1.0-SNAPSHOT-capsule.jar CommandLine Global All
personaal@accessible-serv:~$
```

Figura 32. Visualização dos comandos de crono realizados no servidor

Capítulo 6

Avaliação

Construído o sistema *PIQL*, houve a necessidade de avaliar o sistema. A avaliação do sistema foi realizada através de avaliação por parte de uma perita e através de inquéritos ao público-alvo.

Relativamente à avaliação por perito, a pessoa convidada foi a psicóloga Cristina Godinho, doutorada europeia em Psicologia da Saúde e investigadora¹⁷. A avaliação consistiu em duas reuniões com o objetivo de ter o parecer positivo da investigadora e testar o dinamismo e capacidade do sistema em lidar com os mais diversos tipos de testes, tais como a criação de novos fatores e regras no sistema, diferentes das que foram anteriormente testadas. O domínio utilizado nestes testes foi na área da alimentação, sendo esta completamente distinta da área principal do *PersonAAL* (atividade física).

A primeira reunião foi realizada no dia 5 de Setembro na Faculdade de Ciências da Universidade de Lisboa e teve a presença da psicóloga Cristina Godinho, orientador Carlos Duarte, coorientador José Coelho e o mentorando Diogo Silva. A reunião teve como objetivo informar a psicóloga Cristina Godinho das funcionalidades do sistema *PIQL* e como este funcionava, bem como obter uma ideia do que o sistema necessitava de ter para cumprir com os objetivos estipulados pela investigadora. Foi combinada uma nova reunião e dada uma lista de regras a serem inseridas no sistema juntamente com as mensagens associadas a essas regras (observáveis no Apêndice E – Lista de Regras e Fatores – 1ª Reunião). Juntamente a estes aspetos, foi incentivado a serem realizadas novas funcionalidades no sistema (para cada regra associar uma hora a qual deve ser ativada).

Precedendo a segunda reunião, foi realizada a implementação da nova funcionalidade, juntamente com a lista de regras e mensagens associadas a estas, não se destacando dificuldades de implementação neste processo.

A segunda reunião foi realizada dia 20 de Setembro na Faculdade de Ciências da Universidade de Lisboa que contou com a presença da psicóloga Carolina Godinho, orientador Carlos Duarte, coorientador José Coelho e o mentorando Diogo Silva. A reunião teve como objetivo revelar o funcionamento do sistema com os fatores e regras

¹⁷ Perfil visível em <http://www.cis.iscte-iul.pt/People.aspx?Lang=pt&id=119> , Visitado a 15 de Setembro de 2017

referidas no Apêndice E – Lista de Regras e Fatores – 1ª Reunião. Em termos de execução e processamento do sistema, este foi bem-sucedido, tanto com as regras e mensagens definidas da primeira reunião, bem como pedidos de alteração do programa ao longo da reunião. Após o contacto com o sistema por parte da investigadora, foi acordado que a lista iria ser reformulada (observável em Apêndice F – Lista de Regras e Fatores – 2ª Reunião) e uma nova funcionalidade seria implementada (as regras poderem ser executadas num dia ou em vários dias escolhidos da semana), uma vez que a ideia que a investigadora possuía das funcionalidades e execução do sistema eram diferentes das que o sistema realmente possuía.

Na avaliação ficou provada a facilidade na criação das regras e mensagens definidas pela perita e destacado o dinamismo do sistema, dado que este permitiu incorporar o modelo *HAPA* (modelo utilizado pela perita) sem modificação no código.

Relativamente à avaliação através de inquérito ao público-alvo, este foi realizado na Suíça entre Julho e Agosto de 2017 com 27 participantes (14 do género feminino, 13 do género masculino), com idades compreendidas entre os 55 e os 95 anos, sendo o maior grupo representativo entre os 76 e os 85 anos. Este tipo de avaliação passa pelas perguntas realizadas ao público-alvo do *PIQL*, perguntas essas relacionadas com o sistema e resultados que este fornece, e o grau de interesse do público-alvo nesse sistema. Durante o inquérito mostraram-se três mensagens ao utilizador, cada uma relativa a um tipo de atividade (física, toma de medicação e social):

- “Na última semana quase atingiste o teu objetivo de passos. Não queres andar 10 minutos, de forma a atingires o teu objetivo esta semana?”.
- “Tens que tomar medicação daqui a uma hora. Não te esqueças de tomá-la!”.
- “O tempo está bom hoje, não queres convidar um amigo para dares um passeio?”

Tendo sido fornecidas estas mensagens, foi realizado o questionário (observável na Tabela 12).

Pergunta	Sim	Não
Acha que estas mensagens o podem motivar a ser mais ativo fisicamente?	70%	30%
Acha que estas mensagens o podem ajudar a tomar a sua medicação a horas?	85%	15%
Acha que estas mensagens o podem motivar a ser mais ativo socialmente?	70%	30%
Gostaria de receber este tipo de mensagens?	58%	42%

Tabela 12 - Perguntas realizadas no inquérito a idosos.

Para os que responderam “Sim” à última pergunta, foi perguntado com que frequência desejava receber os avisos, obtendo os resultados da Tabela 13.

Pergunta	Percentagem
1 ou 2 vezes por dia	56%
2 ou 3 vezes por dia	19%
3 ou 4 vezes por dia	6%
2 ou 3 vezes por semana	6%
Não especificaram	13%

Tabela 13 – Respostas à pergunta “Gostaria de receber este tipo de mensagens”

Dos resultados obtidos é possível observar um interesse geral do público-alvo para um sistema semelhante ao *PIQL*. Existe uma resposta positiva acerca das áreas sobre as quais o utilizador gostaria de receber mensagens, semelhantes às que o *PIQL* aborda (presentemente mais focado em atividade física, porém possível os dois outros tipos de atividade). Outro ponto que valida o sistema é que este está desenhado para ser possível prever a frequência com que o utilizador irá receber avisos, sendo possível agradar mais aos utilizadores. Se a maioria dos utilizadores preferir uma ou duas mensagens por dia, pode ser necessário correr as regras definidas numa frequência alta (2 vezes por dia), enquanto caso os utilizadores preferirem 2 ou 3 vezes por semana, as regras podem ser corridas de uma forma menos frequente (dia sim dia não por exemplo).

Capítulo 7

Conclusão

Atendendo ao trabalho realizado ao longo do projeto *PIQL*, é possível afirmar que os objetivos delineados para este projeto foram cumpridos com sucesso:

- Identificar interfaces persuasivas existentes que se possam adequar ao contexto do projeto *PersonAAL*.
- Criação de um módulo/sistema persuasivo através da implementação ou alteração de interfaces persuasivas que se adequem ao contexto do projeto.
- Implementar esse módulo/sistema no projeto *PersonAAL*.

As leituras provenientes do capítulo Trabalho Relacionado permitiram o conhecimento de interfaces com conteúdo persuasivo e após estas serem analisadas, criaram-se bases para a identificação em interfaces persuasivas de aspetos a serem utilizados posteriormente no contexto do *PIQL* (visíveis na Tabela 5 – Aspetos a serem realizados no projeto e Tabela 6 – Técnicas de Alteração de Comportamentos). Relativamente às técnicas de alteração de comportamentos analisadas na leitura supracitada (referidas na Tabela 6), estas foram posteriormente otimizadas através do conhecimento do psicólogo Jan Nordvik, sendo possível afirmar que para além de identificados os aspetos persuasivos necessários para o sistema *PIQL*, estes foram validados por um especialista com conhecimento na área estudada, confirmando o cumprimento do primeiro objetivo.

O segundo objetivo foi cumprido através da criação do sistema *PIQL*, bem como do seu desenho e implementação. Os aspetos persuasivos retirados do capítulo Trabalho Relacionado e do conhecimento fornecido por Jan Nordvik foram inseridos no sistema *PIQL*. A Tabela 14 indica de que forma os aspetos a serem realizados no projeto *PIQL* foram incluídos no seu sistema, onde também é possível encontrar as técnicas de persuasão (referidas no Apêndice C – Lista de Técnicas de Persuasão) que são identificadas no módulo *BCTSelectionFinal* e aplicadas no módulo *BCTInstanceFinal*, originando posteriormente a mensagem persuasiva. A ideia da criação de mensagens com o intuito de melhorar o comportamento do utilizador, bem como a inclusão de módulos persuasivos, tornam o sistema *PIQL* persuasivo, finalizando a ideia que o objetivo foi cumprido.

Descrição dos aspetos a serem realizados no projeto PIQL e sua inclusão no sistema	
Aspetos a serem realizados	Inclusão no sistema PIQL
Através de variáveis antecedentes, prever intenções comportamentais	No módulo <i>IdentifyPatternsFinal</i> são identificados padrões do utilizador, provenientes da sua atividade física.
Não possuir um modelo comportamental que acredite que um indivíduo não pode mudar o seu comportamento de forma radical	Nenhum dos módulos do sistema <i>PIQL</i> ou do seu modelo comportamental (<i>COM-B</i>) impede mudanças bruscas de comportamento por parte do utilizador.
Indivíduos na mesma etapa do comportamento podem ser ajudados da mesma forma	Serão utilizados os mesmos <i>BCT</i> 's para utilizadores com padrões semelhantes de atividades.
Possuir um modelo fácil de programar, dinâmico, que identifique objetivos e seja personalizado para cada utilizador.	O sistema cria padrões, deteta desvios e explora <i>BCT</i> 's únicos para cada utilizador, tornando a mensagem persuasiva única para este. O processo é realizado num sistema sensível à mudança de variáveis (valores de parâmetros e novas regras) com um modelo de comportamento fácil de programar (indicado nas vantagens do modelo)
Contemplar fatores que possam mudar o comportamento (tanto pessoais como exteriores)	Embora virado para atividade física, o sistema <i>PIQL</i> permite a possibilidade de explorar outros fatores (referidos no Apêndice B – Lista de Fatores que afetam comportamento)
Reconhecer tipos de intervenções que devem ser feitas ao utilizador	Sistema possui regras que ativam técnicas a serem executadas no utilizador (módulo <i>BCTSelection</i> seleciona as técnicas, <i>BCTInstanceFinal</i> implementa-as)
Dar as ferramentas ao utilizador para manter o seu comportamento correto	O sistema possui técnicas e mensagens persuasivas caso o utilizador melhore o

	seu comportamento ou mantenha-o correto
Conjugar fatores para tentar apelar a mudança de comportamentos errados	Caso exista uma mudança negativa no comportamento do utilizador, será reconhecido o desvio no padrão e aplicadas técnicas para melhorar o seu comportamento.

Tabela 14 - Descrição dos aspetos a serem realizados no projeto *PIQL* e onde estes estão incluídos no projeto

O terceiro objetivo provém da relação entre os projetos *PIQL* e *PersonAAL*. O uso de módulos existentes no *PersonAAL* (*ContextManager* e *AdaptationEngine*) no sistema *PIQL* é um dos aspetos que permite afirmar que o objetivo foi cumprido. O segundo aspeto é os resultados obtidos nas demonstrações do sistema. Foram realizadas duas demos do projeto *PIQL* com parceiros europeus. A primeira demo foi realizada em 30 de Maio em Oslo com o objetivo de apresentar o sistema com casos exemplo o qual foi considerado positivo pela equipa bem como pelos parceiros europeus e foram realizadas sugestões de forma a melhorar o sistema *PIQL*. A segunda demo foi realizada a 20 de Junho de 2017 em Bruxelas no contexto da revisão *mid-term* do projeto *PersonAAL* com o objetivo de mostrar a evolução do sistema. O sistema funcionou de acordo com o planeado, recebendo a nota “Muito Bom” e elogios dos parceiros europeus.

7.1 Trabalho Futuro

Embora todos estes aspetos sejam bons indicativos de que o sistema *PIQL* está a evoluir na direção correta, existem aspetos que podem ser melhorados, nomeadamente a nível da otimização e funcionalidades. Nesta análise de trabalho futuro não serão abordados aspetos em que o projeto *PersonAAL* precisa de melhorar, porém é necessário entender que há uma forte ligação entre os dois projetos, sendo uma relação simbiótica, onde para haver melhoramentos de um dos projetos, o outro também precisa de evoluir, e há áreas onde o projeto *PersonAAL* precisa de evoluir para que também o sistema *PIQL* possa evoluir.

Tendo este pensamento presente, em termos do projeto *PIQL* um dos aspetos a ser trabalhado no futuro é ao nível da otimização da base de dados. Uma das recomendações passa pela criação de índices. Índices são uma coleção de entradas de registos com o

intuito de encontrar mais rapidamente entradas com um dado par chave-valor. Atendendo ao tipo de procuras realizadas na base de dados, um índice *hash* tem o potencial de tornar as pesquisas mais rápidas, porém atendendo às escritas constantes de registos na base de dados e a consequente atualização dos índices, seria necessário fazer um estudo para observar se a criação de índices seria uma opção a ponderar.

Relativamente à persuasão das mensagens no utilizador, estas poderiam obter melhores resultados caso após a definição de todas as mensagens persuasivas, o sistema *PIQL* criasse um sistema de aprendizagem automática onde fosse avaliada a qualidade de persuasão que cada mensagem iria possuir num dado utilizador numa certa situação, através de um histórico de alterações de comportamento desse utilizador quando sujeito a essa mensagem. Uma mensagem que obteve sucesso em qualquer situação com o utilizador teria uma maior probabilidade de persuasão que uma mensagem que já possuiu um histórico de insucesso.

O sistema *PIQL* possui a capacidade de prever desvios de comportamento através de outros fatores para além da atividade física, sendo que esses ainda não foram desenvolvidos (explicada a razão no capítulo Implementação). Prevê-se que com a evolução do *PersonAAL* e do sistema *PIQL*, seja necessário expandir o tipo de atividades que o sistema analisa, principalmente fatores sociais os quais possuem grande valor persuasivo.

Em suma foi criado o sistema *PIQL* que demonstrou ser capaz de responder às necessidades básicas que lhe eram impostas sempre com uma boa resposta. A avaliação permitiu demonstrar que o sistema é dinâmico o suficiente para ser usado em várias atividades (física e alimentar as mais utilizadas) com diferentes modelos de comportamento. Permitiu também mostrar que este pode ser personalizado de forma a agradar à maioria dos utilizadores, sendo que estes têm uma ideia positiva em relação ao tipo de sistema que o *PIQL* representa. O sistema criado possui uma base sólida, confirmada pelo cumprimento dos objetivos e com perspetivas positivas relativamente à sua evolução.

Bibliografia

- Arroyo, E., Bonanni, L., & Selker, T. (2005). Waterbot: Exploring Feedback and Persuasive Techniques at the Sink. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 631–639. <https://doi.org/10.1145/1054972.1055059>
- Chatterjee, S., & Price, A. (2009). Healthy Living with Persuasive Technologies: Framework, Issues, and Challenges. *Journal of the American Medical Informatics Association*, 16(2), 171–178. <https://doi.org/10.1197/jamia.M2859>
- de Oliveira, R., Cherubini, M., & Oliver, N. (2010). MoviPill. *Proceedings of the 12th ACM International Conference on Ubiquitous Computing - Ubicomp '10*, 251. <https://doi.org/10.1145/1864349.1864371>
- Farinha, C. G. (2012). O MODELO HAPA, FASE INTENCIONAL, NA EXPLICAÇÃO DO ENVOLVIMENTO ESCOLAR. UM ESTUDO EXPLORATÓRIO. *O MODELO HAPA, FASE INTENCIONAL, NA EXPLICAÇÃO DO ENVOLVIMENTO ESCOLAR. UM ESTUDO EXPLORATÓRIO*, 61. <https://doi.org/10.1017/CBO9781107415324.004>
- Fogg, B. (2009). A behavior model for persuasive design. *Proceedings of the 4th International Conference on Persuasive Technology - Persuasive '09*, 1. <https://doi.org/10.1145/1541948.1541999>
- Hkdylru, H. Q. R. I. (2006). Theories of behavior change. *Change*, 1(1990), 1–6. <https://doi.org/10.4236/health.2013.53A078>
- Kok, G. (2014). A practical guide to effective behavior change How to apply theory- and evidence-based behavior change methods in an intervention. *The European Health Psychologist*, 16(5), 156–170.
- Lee, D., Helal, S., Anton, S., De Deugd, S., & Smith, A. (2012). Participatory and persuasive telehealth. *Gerontology*, 58(3), 269–281. <https://doi.org/10.1159/000329892>
- Meyer, E. S., Tran, T., & Greenwood, M. (2016). Statistical methods for detecting groups of patterns in daily step count activity profiles. *Skyline - The Big Sky Undergraduate Journal*, 4(1), 276–290. <https://doi.org/10.1145/2967934.2968088>
- Michie, S., Richardson, M., Johnston, M., Abraham, C., Francis, J., Hardeman, W., ... Wood, C. E. (2013). The behavior change technique taxonomy (v1) of 93 hierarchically clustered techniques: Building an international consensus for the reporting of behavior change interventions. *Annals of Behavioral Medicine*, 46(1), 81–95. <https://doi.org/10.1007/s12160-013-9486-6>
- Morris, J., Marzano, M., Dandy, N., & O'Brien, L. (2012). Theories and models of behaviour and behaviour change. *Forestry, Sustainable Behaviours and Behaviour Change: Theories*, 1–27.
- Nawyn, J., Intille, S. S., & Larson, K. (2006). Embedding Behavior Modification Strategies into a Consumer Electronic Device: A Case Study. *UbiComp 2006: Ubiquitous Computing, LNCS 4206*, 4206(19), 297–314. <https://doi.org/10.1073/pnas.1105195108>
- Raffle, H., Go, J., Spasojevic, M., Revell, G., Mori, K., Ballagas, R., ... Freed, N. (2011). Hello, is grandma there? let's read! StoryVisit. *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems - CHI '11*, 1195. <https://doi.org/10.1145/1978942.1979121>
- S.Sutton. (2002). Health Behavior - Psychosocial Theories. *University of Cambridge*,

10. <https://doi.org/10.1016/B978-0-08-097086-8.14153-4>
- Schwarzer, R. (2008). Modeling health behavior change: How to predict and modify the adoption and maintenance of health behaviors. *Applied Psychology*, 57(1), 1–29. <https://doi.org/10.1111/j.1464-0597.2007.00325.x>
- Swamy Ananthanarayan, S., & Siek, K. a. (2012). Persuasive Wearable Technology Design for Health and Wellness. *Proceedings of the 6th International Conference on Pervasive Computing Technologies for Healthcare*, 236–240. <https://doi.org/10.4108/icst.pervasivehealth.2012.248694>
- UCL Working Group. (2015). PWP Training Review: Behaviour change models and strategies relevant to Psychological Wellbeing Practitioners. *Under the Aegis of the National IAPT Team/ NHS England/ Department of Health*, (March), 1–14.

Apêndice

Apêndice A – Ficheiro Excel de Atividade Física

1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	10	0	0	0	0	0	139	33	62
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0	0	0	19	157	68	331
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	52
0	0	0	0	0	0	0	0	0	0	0	69	526	68
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	153
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	7	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	7	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	5	0	0	7	4	0	0	0	0	0	8	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	9	0	382	806	514	94	36	110	0
0	0	0	0	9	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	15	0	0	0
0	0	0	0	0	0	0	0	0	0	0	54	74	166
0	0	0	0	0	8	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	65	52	65	209
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	8	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	4	0	0	0	5	103	70	186
85	86	87	88	89	90	91	92	93	94	95	96 Day	Group	
59	0	0	0	0	0	0	0	0	0	0	0	2 Summer	
0	0	0	0	0	0	0	0	0	0	0	0	3 Summer	
0	0	0	0	0	0	0	0	0	0	0	0	4 Summer	
54	0	0	0	0	0	0	0	0	0	0	0	5 Summer.W	
0	0	0	0	0	0	0	0	0	0	0	0	6 Summer.W	
0	0	0	0	0	0	0	0	0	0	0	0	7 Summer	
0	0	0	0	0	0	0	0	0	0	0	0	8 Summer	
0	0	0	0	0	0	0	0	0	0	0	0	9 Summer	
0	0	0	0	0	0	0	0	0	0	0	0	10 Summer	
0	0	0	0	0	0	0	0	0	0	0	0	11 Summer	
0	0	0	0	12	0	0	0	0	0	0	0	14 Summer	
0	0	0	0	0	0	0	0	0	0	0	0	15 Summer	
0	0	0	0	0	0	0	0	0	0	0	0	16 Summer	
0	0	0	0	0	0	0	0	0	0	0	0	17 Summer	
0	0	0	0	0	0	0	0	0	0	0	0	18 Summer	
0	0	0	0	0	0	0	0	0	0	0	0	19 Summer.W	
24	116	73	0	0	0	0	0	0	0	0	0	20 Summer.W	
5	106	94	25	419	0	16	0	0	0	0	0	21 Summer	
0	0	0	0	0	0	0	0	0	6	0	0	22 Summer	
28	0	12	372	114	14	273	0	0	0	0	0	23 Summer	
0	0	0	0	0	0	0	0	0	0	0	0	24 Summer	
0	0	0	0	0	0	0	0	0	0	0	0	25 Summer	
0	0	0	0	0	0	0	0	0	0	0	0	26 Summer.W	
0	0	0	8	0	8	0	0	0	0	0	0	27 Summer.W	
0	0	0	0	0	0	0	0	0	0	0	0	29 School	
0	0	0	0	8	0	0	0	0	0	0	0	30 School	
44	0	0	0	0	0	0	0	0	0	0	0	31 School	
0	0	0	0	0	0	0	0	0	0	0	0	32 School	
0	0	0	0	0	0	10	0	0	0	0	0	33 School.W	
108	7	0	0	0	0	0	0	0	0	0	0	34 School.W	
7	0	0	0	0	0	0	0	0	0	0	0	35 School	
0	0	0	0	0	0	0	0	0	0	0	0	36 School	

Apêndice B – Lista de Fatores que afetam comportamento

PHYSICAL ACTIVITY	
CAPABILITY	Effect on behaviour
High BMI (Body Mass Index)	Negative
Hungry/not been eating	Negative
Illnesses incl. cognitive impairment	Negative
Injuries	Negative
Low mood (Sadness/depression/loneliness)	Negative
Pain	Negative
Smoking	Negative
Low Sleep Quality	Negative
Unfit (number of steps and walking speed at baseline?)	Negative
Weak grip strength	Negative
Fit (number of steps and walking speed at baseline?)	Positive
Happiness/positive mood	Positive
Knows how to perform the activity	Positive
Normal grip strength	Positive
Self-efficacy	Positive
Walking aids	Positive
Normal BMI (Body Mass Index)	Positive
OPPORTUNITY	
Activity far away from home	Negative
Lack equipment needed for the activity	Negative
Lack of guidance	Negative
Activity nearby home	Positive
Affordable activity	Positive
Company (practical support)	Positive
Got equipment needed for the activity	Positive
Local safety	Positive
Local weather	Positive
Free Agenda	Positive

Schedule Activity	Positive
Busy Agenda	Negative
Finishing Eating	Negative
Need of transportation	Negative
Local Weather	Negative
MOTIVATION	
Below step goal	Negative
Negative memories from activity	Negative
Over step goal	Positive
Company (social)	Positive
Impress caretaker	Positive
Interested in the activity	Positive
Introjected regulation (feelings of guilt or shame)	Positive
Perceived physical benefits	Positive
Perceived psychological benefits (well-being)	Positive
Perceived social benefits	Positive
Positive feelings and memories about the activity	Positive

SOCIAL ACTIVITY	
CAPABILITY	Effect on behaviour
High physical function	Positive
High social media skills	Positive
High smart phone skills	Positive
High iPad/tablet skills	Positive
High Computer skills	Positive
High Communication skills	Positive
Happiness/positive mood	Positive
High EQ (Social Skills)	Positive
High Self-efficacy	Positive
Low physical function	Negative
Low social media skills	Negative

Low smart phone skills	Negative
Low iPad/tablet skills	Negative
Low Computer skills	Negative
Low Communication skills	Negative
Hearing impairment	Negative
Severe vision impairment	Negative
Low Sleep Quality	Negative
Walking aids	Negative
Low EQ (Social Skills)	Negative
Sadness/negative mood	Negative
Low Self-efficacy	Negative
OPPORTUNITY	
Spouse	Positive
Children	Positive
Grandchildren	Positive
Close relatives / friends	Positive
Smart phone	Positive
iPad/tablet	Positive
PC	Positive
Memberships in organisations/clubs	Positive
Local safety	Positive
Local weather	Positive
Time (no conflicting appointments)	Positive
Number of friends	Positive
Number of acquaintances	Positive
Number of interests/hobbies	Positive
Total size of social network (family, friends and acquaintances)	Positive
Need of transportation	Negative

Distance (kilometers) to family and friends	Negative
No Spouse	Negative
No Children	Negative
No Grandchildren	Negative
No Close relatives / friends	Negative
No Smart phone	Negative
No iPad/tablet	Negative
No PC	Negative
No Memberships in organisations/clubs	Negative
MOTIVATION	
Very socially active	Positive
Extrovertness	Positive
Impress caretaker	Positive
Interested in the activity	Positive
Introjected regulation (feelings of guilt or shame)	Positive
Perceived physical benefits	Positive
Perceived psychological benefits (well-being)	Positive
Perceived social benefits	Positive
Positive feelings and memories about the activity	Positive
Low social activity	Negative
Negative memories from activity	Negative
Introvertness	Negative

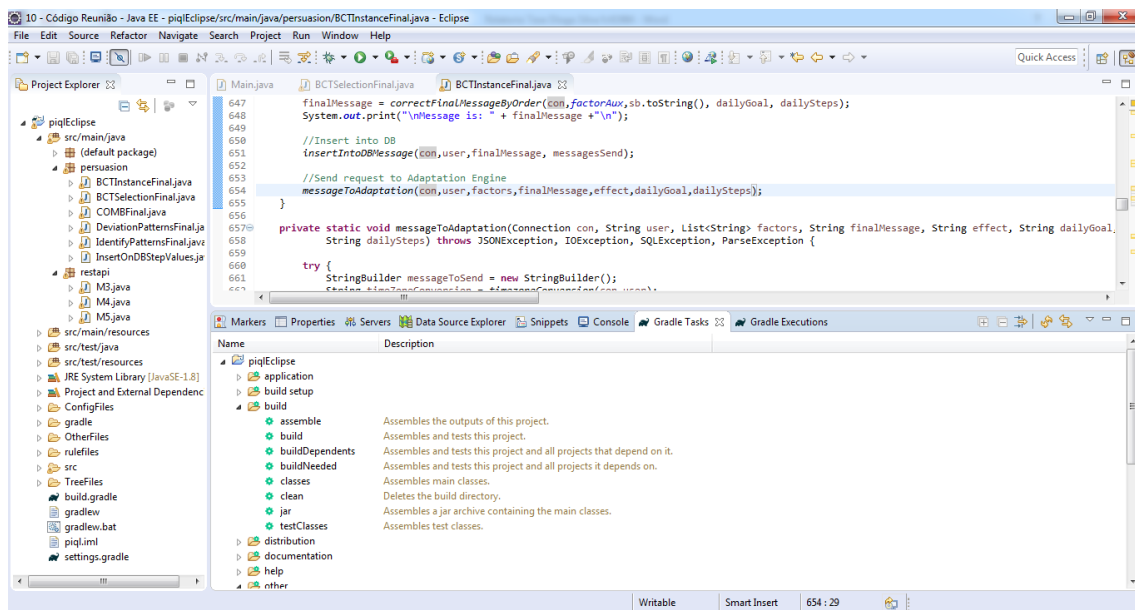
Apêndice C – Lista de Técnicas de Persuasão

Identity	Social	Rational (values/beliefs)	Positive emotional reinforcement	Negative emotional reinforcement	Problem solving
----------	--------	------------------------------	-------------------------------------	-------------------------------------	--------------------

Apêndice D – Criação de cápsulas

Passo 1 – Necessário possuir o ambiente de desenvolvimento Eclipse, e através do *Eclipse MarketPlace*, instalar o *plugin BuildShip*.

Passo 2 – Importar o projeto como projeto *Gradle*. Neste momento o estado do projeto será este:



Passo 3 – Editar o ficheiro *build.gradle*. Neste deve ser introduzido os plugins, repositórios, dependências e para criar a cápsula é necessário fazer o que está inserido na figura seguinte. Task *serverCapsule* cria o nome da tarefa que vai criar a capsula, *applicationClass* precisa do nome da classe e *baseName* indica como se vai chamar a cápsula.

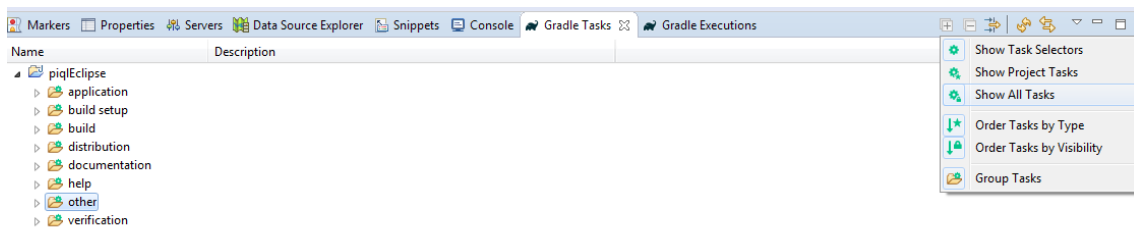

```

1 plugins {
2     id "us.kirchmeier.capsule" version "1.0.2"
3 }
4
5 group 'piql'
6 version '1.0-SNAPSHOT'
7
8 apply plugin: 'java'
9 apply plugin: 'application'
10
11 mainClassName = 'Main'
12
13 sourceCompatibility = 1.8
14
15 repositories {
16     mavenCentral()
17 }
18
19 dependencies {
20     compile group: 'org.rosuda.REngine', name: 'Rserve', version: '1.8.1'
21     compile group: 'org.apache.httpcomponents', name: 'httpclient', version: '4.5.3'
22     compile group: 'org.json', name: 'json', version: '20170516'
23     compile group: 'commons-io', name: 'commons-io', version: '2.5'
24     compile group: 'org.eclipse.jetty', name: 'jetty-servlet', version: '9.4.5.v20170502'
25     compile group: 'org.glassfish.jersey.containers', name: 'jersey-container-jetty-servlet', version: '2.25.1'
26     compile group: 'org.apache.poi', name: 'poi', version: '3.16'
27     compile group: 'org.apache.poi', name: 'poi-ooxml', version: '3.16'
28     compile group: 'mysql', name: 'mysql-connector-java', version: '5.1.42'
29 }
30
31 task serverCapsule(type: MavenCapsule) {
32     applicationClass 'Main'
33     baseName 'piqlServer'
34 }

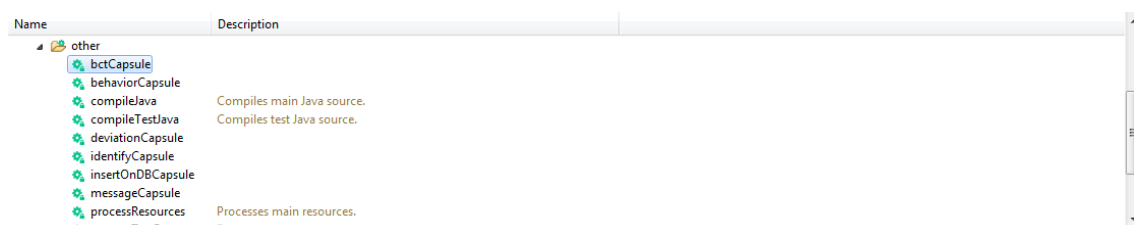
```

Passo 4 – Como possível visualizar na imagem do passo 2, é necessário ir ao tab “Gradle Tasks”. Caso o projeto não tenha erros de compilação, é necessário ir à pasta build, e carregar no botão build, de forma a construir o projeto.

Passo 5 – Com o projeto nesta etapa construído, é possível pedir a execução das cápsulas. Caso não apareça a pasta Other na tab “Gradle tasks”, é necessário carregar essa opção. A figura seguinte indica o sítio:



Passo 6 – Indicar a cápsula que deve ser compilada



Passo 7 – Aguardar a compilação da cápsula

IF *[Actual consumption]* < *[Recommendation]* AND *[Intention]* <=
[Recommendation] then <P.T. Knowledge + Positive feedback (1) + Benefits + Social
norms + Self-efficacy >

IF *[Actual consumption]* < *[Recommendation]* AND *[Intention]* >=
[Recommendation] then <P.T. Goal setting + Planning + Coping Planning (1) + Self-
monitoring + Positive feedback (2)>

IF *[Actual consumption]* > *[Recommendation]* then <P.T. Coping Planning (2) +
Self-monitoring + Positive feedback (3) + Reinforcement >

Behavior change techniques

- Knowledge
 - Sabia que deve comer pelo menos 5 porções de frutas e vegetais por dia?
 - Uma porção de fruta é o equivalente a uma maçã, laranja, uma talhada de melão ou uma mão cheia de morangos e sumos naturais 100% fruta
 - Uma porção de vegetais é o equivalente a uma salada, uma sopa, ou um acompanhamento de vegetais
- Positive feedback (1)
 - Atualmente come XX porções de frutas e vegetais por dia. Basta juntar mais XX porção(ões) para chegar ao número mágico das 5!
- Benefits
 - Comer pelo menos 5 porções diárias de frutas e vegetais ajuda-o a prevenir diversas doenças, como cancro, diabetes e doenças cardiovasculares
 - Comer frutas e vegetais diariamente ajuda-o a manter-se em forma e a sentir-se bem consigo próprio(a)
- Social norms
 - A maioria dos (Lisboetas) gostaria de comer mais frutas e vegetais por dia. E você?
 - XXX (dados sobre consumo)
- Self-efficacy
 - É fácil comer 5 porções diárias de frutas e vegetais! Experimente começar sempre as refeições principais com sopa e juntar um acompanhamento de vegetais ou salada e vai de certeza conseguir!
- Goal setting

- Vamos estabelecer o seu objetivo para esta semana. Quantas porções de frutas e vegetais gostaria de passar a comer por dia?
- Planning
 - Em que momento do seu dia seria possível aumentar o seu consumo de fruta? Por exemplo, ao pequeno-almoço, no intervalo da manhã, nas refeições principais, ao lanche?
 - Como é que seria possível aumentar o seu consumo de vegetais? Por exemplo, comer sopa à refeição, acompanhar sempre a refeição com vegetais ou salada, preferir snacks à base de vegetais?
- Coping Planning (1)
 - Pense por um momento... O que o impede de conseguir comer 5 porções de frutas e vegetais por dia? (XXX)
 - Como evitar XXX de modo a conseguir atingir os seus objetivos?
- Self-monitoring
 - Pense no seu dia e no que comeu em todas as refeições que fez. Quantas porções de frutas e vegetais comeu hoje?
- Positive Feedback (2)
 - Parabéns! Esta semana conseguiu atingir o seu objectivo! // Parabéns! Hoje conseguiu atingir a recomendação da OMS relativa ao consumo de frutas e vegetais.
 - Está no bom caminho... Esta semana conseguiu atingir % do seu objectivo (mostrando informação sobre o objectivo e nível atingido – i.e., média de porções diárias)
- Coping Planning (2)
 - Pense por um momento... O que poderia impedi-lo de conseguir comer 5 porções de frutas e vegetais por dia? (XXX)
 - Como evitar XXX de modo a conseguir atingir os seus objetivos?
- Self-monitoring
 - Pense no seu dia e no que comeu em todas as refeições que fez. Quantas porções de frutas e vegetais comeu hoje?
- Positive feedback (3)
 - Parabéns! Hoje / esta semana conseguiu comer mais de 5 porções de frutas e vegetais por dia!

- Reinforcement
 - Ganhou XXX pontos e o seu bem-estar agradece!

Apêndice F – Lista de Regras e Fatores – 2ª Reunião

Background information

Measured every day:

- Actual consumption (2 items)
- Intention (2 items)

Rationale

(based on the Health Action Process Approach, Schwarzer 2008)

Every day:

IF *[Actual consumption]* < *[Recommendation]* then <P.T. Positive Feedback (1)>

IF *[Actual consumption]* >= *[Recommendation]* then <P.T. Positive Feedback (2) + Reinforcement (1)>

Once a week (one technique per day):

IF *[Actual consumption]* < *[Recommendation]* AND *[Intention]* < *[Recommendation]* then <P.T. Knowledge + Monitoring of behavior + Benefits + Social norms + Self-efficacy (1)>

IF *[Actual consumption]* < *[Recommendation]* AND *[Intention]* >= *[Recommendation]* then <P.T. Monitoring of behavior + Goal setting + Planning + Coping Planning (1) + Self-efficacy (2)>

IF *[Actual consumption]* > *[Recommendation]* then <P.T. Coping Planning (2) + Self-efficacy (3) + Monitoring of behavior + Reinforcement (2)>

Behavior change techniques

Diariamente:

- Positive feedback (1)
 - Ontem comeu XX porções de frutas e vegetais. Está no bom caminho! Se hoje comer mais XX porção(ões) conseguirá chegar ao número mágico das 5!
- Positive Feedback (2)
 - Parabéns! Ontem conseguiu comer 5 ou mais porções de frutas e vegetais!
- Reinforcement
 - Ganhou XXX pontos e o seu bem-estar agradece!

Semanalmente:

- Knowledge
 - Sabia que deve comer pelo menos 5 porções de frutas e vegetais por dia?
 - Uma porção de fruta é o equivalente a uma maçã, laranja, uma talhada de melão ou uma mão cheia de morangos e sumos naturais 100% fruta
 - Uma porção de vegetais é o equivalente a uma salada, uma sopa, ou um acompanhamento de vegetais
- Monitoring of behavior
 - Na semana passada comeu em média XX frutas e vegetais por dia.
- Benefits
 - Comer pelo menos 5 porções diárias de frutas e vegetais ajuda-o a prevenir diversas doenças, como cancro, diabetes e doenças cardiovasculares
 - Comer frutas e vegetais diariamente ajuda-o a manter-se em forma e a sentir-se bem consigo próprio(a)
- Social norms
 - A maioria dos (Lisboetas) gostaria de comer mais frutas e vegetais por dia. E você?
 - XXX (dados sobre consumo)

- Self-efficacy (1)
 - É fácil comer 5 porções diárias de frutas e vegetais! Experimente começar sempre as refeições principais com sopa e juntar um acompanhamento de vegetais ou salada e vai de certeza conseguir!
- Goal setting
 - Vamos estabelecer o seu objetivo para esta semana. Quantas porções de frutas e vegetais gostaria de passar a comer por dia?
- Planning
 - Em que momento do seu dia seria possível aumentar o seu consumo de fruta? Por exemplo, ao pequeno-almoço, no intervalo da manhã, nas refeições principais, ao lanche?
 - Como é que seria possível aumentar o seu consumo de vegetais? Por exemplo, comer sopa à refeição, acompanhar sempre a refeição com vegetais ou salada, preferir snacks à base de vegetais?
- Coping Planning (1)
 - Pense por um momento... O que o impede de conseguir comer 5 porções de frutas e vegetais por dia? (XXX)
 - Como evitar XXX de modo a conseguir atingir os seus objetivos?
- Self-efficacy (2)
 - Está no bom caminho! Planeando a mudança vai ser mais fácil conseguir atingir o seu objectivo.
- Coping Planning (2)
 - Pense por um momento... O que poderia impedi-lo de conseguir comer 5 porções de frutas e vegetais por dia? (XXX)
 - Como evitar XXX de modo a conseguir atingir os seus objetivos?
- Self-efficacy (3)
 - Mais difícil do que mudar é manter essa mudança. Mas não desanime! Um dia não são dias, vai conseguir continuar no bom caminho!
- Reinforcement (2)
 - Parabéns! Esta semana conseguiu atingir a recomendação da OMS relativa ao número de porções diárias de frutas e vegetais!